INTERNET DE LAS COSAS

Teoría y práctica

Armando Román Gallardo, José Román Herrera Morales, Sara Sandoval Carrillo,
María Andrade Aréchiga, Erika Margarita Ramos Michel

Coordinadores

UNIVERSIDAD DE COLIMA

Internet de las cosas

Teoría y práctica

enfoque académico

Universidad de Colima
Dr. Christian Jorge Torres Ortiz Zermeño, Rector
Mtro. Joel Nino Jr., Secretario General
Mtra. Vianey Amezcua Barajas, Coordinadora General de Comunicación Social
Mtra. Gloria Guillermina Araiza Torres, Directora General de Publicaciones

Internet de las cosas
Teoría y práctica

Armando Román Gallardo
José Román Herrera Morales
Sara Sandoval Carrillo
María Andrade Aréchiga
Erika Margarita Ramos Michel
Coordinadores

UNIVERSIDAD DE COLIMA

© Universidad de Colima, 2023
Avenida Universidad 333
C.P 28040, Colima, Colima, México
Dirección General de Publicaciones
Teléfonos: 312 316 1081 y 312 316 1000, extensión: 35004
Correo electrónico: publicaciones@ucol.mx
http://www.ucol.mx

Derechos reservados conforme a la ley
Publicado en México / *Published in Mexico*

ISBN eBook: 978-607-8814-44-2
DOI: 10.53897/LI.2023.0001.UCOL

Este libro está bajo la licencia de Creative Commons, Atribución – NoComercial – CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0).

Usted es libre de: Compartir: copiar y redistribuir el material en cualquier medio o formato. Adaptar: remezclar, transformar y construir a partir del material bajo los siguientes términos: Atribución: Usted debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo de la licenciante. NoComercial: Usted no puede hacer uso del material con propósitos comerciales. CompartirIgual: Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original.

This work is licensed under a Creative Commons Attribution – NonCommercial – ShareAlike 4.0 International License.

You are free to: Share: copy and redistribute the material in any medium or format. Adapt: remix, transform, and build upon the material under the following terms: Attribution: You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. NonCommercial: You may not use the material for commercial purposes. ShareAlike: If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

Proceso editorial certificado con normas ISO desde 2005
Dictaminación y edición registradas en el Sistema Editorial Electrónico PRED

Registro: LI-031-22
Recibido: Octubre de 2022
Publicado: Enero de 2023

Índice

Introducción general 7

Capítulo 1
Introducción al Internet de las cosas 9
Carlos Alberto Flores Cortés
María Andrade Aréchiga
Ricardo Acosta Díaz
Juan Manuel Ramírez Alcaraz

Capítulo 2
Ecosistemas de IoT 25
José Román Herrera Morales
Sara Sandoval Carrillo
Armando Román Gallardo

Capítulo 3
Componentes principales de *hardware* para IoT 39
Armando Román Gallardo
Carlos Alberto Flores Cortés
José Román Herrera Morales
Omar Álvarez Cárdenas

Capítulo 4
Conectividad IoT 57
Juan Manuel Ramírez Alcaraz
María Guadalupe Álvarez Negrete
Arturo Cano Rueda
Carlos Alberto Flores Cortés

Capítulo 5
Protocolos de comunicación de los dispositivos IoT 83
Omar Álvarez Cárdenas
Margarita Glenda Mayoral Baldivia
Raúl T. Aquino Santos

Capítulo 6
Plataformas para IoT 115
José Román Herrera Morales
Armando Román Gallardo
Sara Sandoval Carrillo
Margarita Glenda Mayoral Baldivia

Capítulo 7
Aplicaciones del aprendizaje automático para el Internet de las cosas ... 131
Erika M. Ramos Michel
María Andrade Aréchiga
Jorge Rafael Gutiérrez Pulido
Pedro Damián Reyes
Ricardo Acosta Díaz

Capítulo 8
Ejercicios prácticos de IoT 153
Armando Román Gallardo
José Román Herrera Morales
Sara Sandoval Carrillo

Introducción general

El concepto de Internet de las cosas (IoT, por sus siglas en inglés) hace referencia a una serie de dispositivos conectados a la red, así como a las tecnologías que facilitan la comunicación entre estos a través de la nube o entre ellos mismos. Esto es viable, ya que actualmente la electrónica que lo hace posible mediante los *chips* y placas de desarrollo ha bajado de precio significativamente, y la infraestructura de las redes de telecomunicaciones con grandes anchos de banda que avanzan considerablemente ha permitido que millones de estos dispositivos estén conectados a través de Internet. La conexión de objetos cotidianos como un cepillo de dientes, vehículos, aspiradora de la casa y gran variedad de máquinas, pueden utilizar sus sensores para recopilar gran cantidad de información y, por tanto, procesarla con algoritmos inteligentes, como los de aprendizaje automático, para atender de manera inteligente las necesidades de los usuarios.

Son varias las tecnologías que se utilizan en IoT, entre ellas se pueden mencionar la computación de borde para mejorar la velocidad con que se transmite la información, reduciendo la latencia en las comunicaciones; la computación en la nube, que brinda la posibilidad de almacenar grandes cantidades de información y la administración de los dispositivos de manera remota, además de poner las grandes cantidades de datos a disposición de los dispositivos conectados y aplicaciones de *software* para la toma de decisiones de forma inteligente, haciendo uso de algoritmos de *Machine Learning* que identifican patrones en datos masivos y elaboran predicciones en tiempo real.

Es importante incluir las habilidades en IoT en el currículo de los programas de ingeniería. En el caso de la Universidad de Colima, a través de la Facultad de Telemática, se cuenta con varios

programas de licenciatura y posgrado que incluyen asignaturas sobre IoT, por lo que es necesario tener un panorama general y particular sobre esta área, que es el propósito principal de este libro.

En el capítulo 1 se hace una introducción de la tecnología IoT, al Internet de todo, al Internet industrial de las cosas, y se muestran aplicaciones en distintas áreas. En el capítulo 2 se presentan diferentes perspectivas para definir los ecosistemas de IoT y sus aplicaciones en diferentes sectores. Los componentes principales de *hardware* que se requieren para el desarrollo de proyectos y aplicaciones en IoT son descritos en el capítulo 3. En el capítulo posterior se explican las tecnologías de comunicación que permiten conectar los dispositivos de IoT que están relacionadas con este, así como la conectividad necesaria y suficiente para que se puedan comunicar.

Los distintos protocolos de comunicación que brindan diferentes formas de enviar y recibir información en el menor tiempo posible se presentan en el capítulo 5. En el capítulo 6 se definen las características de las plataformas de IoT y cómo se clasifican según la orientación de los servicios que ofrecen. En el capítulo 7 se muestra una breve descripción tanto histórica como actual de conceptos relacionados con IoT y aplicaciones relevantes de aprendizaje automático. Se cierra con la presentación de ejercicios prácticos de IoT en el capítulo 8, cuya realización permite interactuar con la electrónica, la conectividad y los protocolos de comunicación para distintas tareas específicas de estos dispositivos.

Capítulo 1

Introducción al Internet de las cosas

Carlos Alberto Flores Cortés
María Andrade Aréchiga
Ricardo Acosta Díaz
Juan Manuel Ramírez Alcaraz

Resumen

Amás de 20 años de que Kevin Ashton introdujera el término *el Internet de las cosas*, es relevante hacer una revisión del estado del arte, la situación actual y el futuro de lo que hoy en día significa esta creciente industria. Por lo cual, en este apartado se presenta una breve semblanza de su evolución, aplicaciones, oportunidades y retos.

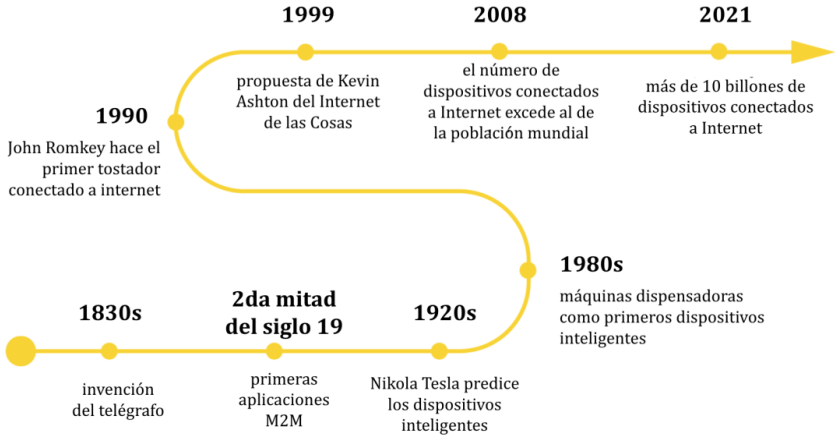
El Internet de las cosas

El concepto inicial presentado por Kevin Ashton se basa en agregar tecnología (RFID) en la cadena de suministro de la empresa Procter & Gamble, para de esta forma hacer posible que los objetos o *cosas* fueran capaces de generar o entregar información de manera directa, eliminando o sustituyendo la participación de las personas (Ashton, 2015). Para esa época, este cambio en el enfoque de alimentar a los sistemas, aunque pareciera simple, en realidad era muy revolucionario pues, hasta ese entonces, prácticamente toda la información con la que se alimentaba a los sistemas provenía o era suministrada por personas.

Diez años más tarde, Ashton y otros investigadores como Chui, Loffer y Roberts (2010), haciendo una revisión de los pocos avances que hasta ese entonces se tenían, anticipaban un cambio importante en las rutas que daban origen o generaban información en donde el mundo físico se convertiría en el principal proveedor de información. Este cambio estaría caracterizado por tres componentes principales: 1) la generación de grandes volúmenes de datos, 2) el desarrollo de tecnologías para el almacenamiento, procesamiento y análisis de esos datos y 3) el funcionamiento autónomo, es decir, sin intervención humana.

Hoy en día, cualquier dispositivo con capacidad para conectarse a otro es considerado IoT. Esta industria representa a nivel mundial un mercado de $742 billones de dólares y se estima en más de 10 billones el número de dispositivos IoT activos (Fernández, 2022). Y aunque estos números ya son impresionantes, esta industria aún tiene mucho potencial de crecimiento pues algunos estudios estiman que para el 2030 el número de dispositivos conectados superará los 25 billones y un valor económico de entre 4 y 5 trillones de dólares para 2025 (Fernández, 2022). En la figura 1, se presenta un resumen de acontecimientos históricos que han sido relevantes para el desarrollo del Internet de las cosas, desde la invención del telégrafo en los años 1830 hasta la actualidad, con el importante número de dispositivos conectados a Internet (AVSystem, 2021).

Figura 1. Eventos relevantes relacionados con la evolución del Internet de las cosas



Fuente: Elaboración propia, adaptado de AVSystem (2021).

Entre los principales detonantes del crecimiento y desarrollo del Internet de las cosas se pueden destacar:

- *La conectividad ubicua.* La disponibilidad de tecnología para realizar conectividad omnipresente o ubicua a bajo costo y altas velocidades, lo que permite que casi cualquier cosa pueda conectarse a Internet.
- *La amplia adopción de redes de datos basadas en el protocolo IP.* El protocolo IP se ha convertido en el estándar para la gran mayoría de las redes de datos y aplicaciones de *software* a nivel mundial, haciendo que su incorporación en una amplia gama de dispositivos sea fácil y a bajo costo.
- *La industria del cómputo*. El constante incremento en el desarrollo de dispositivos con mayores capacidades de procesamiento a precios más bajos y con bajos niveles de consumo de energía.
- *La miniaturización de componentes.* Los avances tecnológicos en la fabricación de componentes de cómputo y comunicación de menor tamaño permiten que este

tipo de componentes puedan incorporarse en objetos cada vez más pequeños.

- *Los avances en el análisis de datos*. El desarrollo de nuevos algoritmos y el incremento en la capacidad de cómputo, almacenamiento de datos y servicios en la nube hacen posible el análisis de grandes volúmenes de datos, así como la extracción de información y conocimiento.
- *El surgimiento del c*ómputo en la *nube.* La disponibilidad de recursos de cómputo de manera remota, así como la administración y almacenamiento de información permiten a dispositivos pequeños acceder de manera remota a recursos con grandes capacidades de procesamiento y analíticas.

El Internet de todo

El Internet de todo (IoE por sus siglas en inglés), es un término que algunos autores utilizan al referirse a aplicaciones o proyectos de investigación relacionados con el IoT y tiene que ver con una versión actualizada para hacer referencia al IoT por la gran cantidad de avances que se han hecho desde su nacimiento como concepto. Tomando en cuenta que la definición original del IoT fue en su momento un concepto muy novedoso y visionario, hoy en día es un concepto cuya visión y alcance, aunque en sus principios básicos sigue siendo vigente, han sido completamente rebasadas. Langley et al. (2021) hacen un análisis de la forma en la que no solo objetos comunes como timbres de puertas, refrigeradores, televisores, carros, sino también otros que quizá no imaginábamos, han evolucionado para incorporar nuevas capacidades que tienen que ver con la conectividad a la red, el uso de sensores y sistemas de control.

Los autores hacen notar una explosión de la conectividad que hoy en día pasa en la mayoría de los casos como desapercibida y destacan a la hiperconectividad como un concepto clave para la transición del IoT al IoE. En este contexto, también describen hiperconectividad como un enorme número de mecanismos de comunicación e interacción que están siempre activos, son de fácil

acceso y ricos en información e interacciones (Langley et al, 2021). Así, el IoE extiende el concepto de IoT al agregar enlaces a datos, personas y procesos (negocios).

El Internet industrial de las cosas

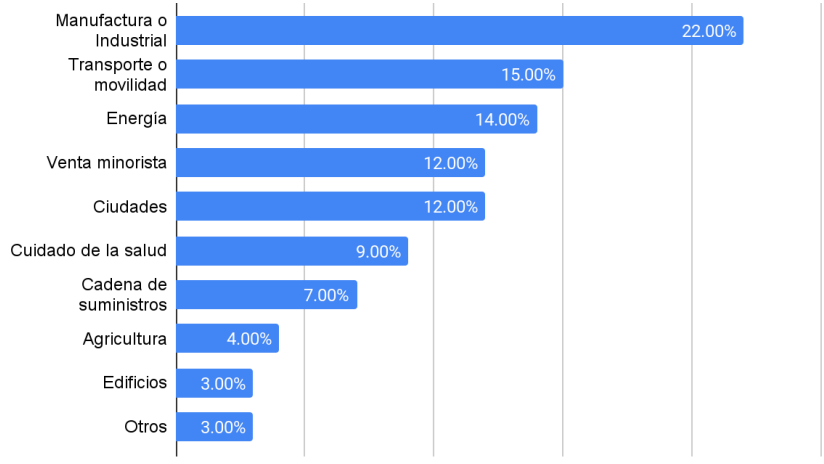
De forma similar al IoE, el I-IoT (del inglés Industrial-Internet of Things) se refiere a un subconjunto o especialización del IoT (Veneri y Capasso, 2018). No hace referencia a una nueva tecnología en particular sino a todo el conjunto de valor de un producto. Así, el I-IoT tiene impacto en todos los sectores del mundo industrial modificando los procesos en cada etapa del ciclo de vida de un producto, incluyendo la forma en que es diseñado, elaborado, entregado, vendido y recibe mantenimiento. Dentro de los casos de uso destacan diferentes tipos de operaciones de manufactura, administración de activos, monitoreo y operación remota, mantenimiento basado en ciertas condiciones y *Big Data*. En términos generales, como su nombre lo indica, I-IoT se refiere estrictamente a todo lo relacionado con la industria y las siguientes son algunas de sus principales características:

- Utilización de ciber-seguridad como un tema crítico en las soluciones que se desarrollan.
- Incorporación de mecanismos para garantizar que los dispositivos digitales funcionen de manera ininterrumpida.
- Capacidad para coexistir en ambientes en donde también existen sistemas o tecnologías legadas.
- Utilización de redes especializadas y determinísticas con capacidades para soportar decenas de miles de controladores, robots y maquinaria.
- Son objetos físicos que, al compararlos con aquellos fabricados para uso comercial, son mucho más complejos en términos de sus características y capacidades.
- En su fabricación se pone mucha atención a factores como la resiliencia y disponibilidad (en términos de su operación o funcionamiento y robustez).
- Protección intelectual sobre su diseño.

Aplicaciones

Los proyectos relacionados con el IoT que a la fecha se han desarrollado atienden o se orientan a diferentes sectores o industrias. En 2020, la empresa IoT Analytics publicó los resultados de un estudio que realizó, en donde se consideraron un total de 1,414 proyectos (Lasse Lueth, 2020). Como se muestra en la figura 2, los proyectos que se analizaron fueron clasificados en 10 áreas de aplicación.

Figura 2. Clasificación de proyectos en las áreas de aplicación



Fuente: Adaptado de Lasse Lueth (2020).

A continuación, se presenta de manera resumida en qué aspectos se han centrado los desarrollos en las cinco primeras categorías de la clasificación de proyectos presentada en la Figura 2, así como algunos ejemplos de los productos que ya se encuentran disponibles en el mercado.

Manufactura o industrial

En el sector industrial, las aplicaciones desarrolladas se vinculan al crecimiento de la Industria 4.0 y dentro de sus propósitos destacan el maximizar la utilización de los activos, aumentar la eficiencia en los procesos relacionados con el desarrollo de productos, la

optimización de las cadenas de suministro y la prevención de fallas en los equipos. Las aplicaciones agrupadas en esta área son muy amplias e incluyen sectores como el automotriz, vehículos comerciales, aeroespacial, por mencionar algunos. Por ejemplo, la empresa Howden, con la participación de Microsoft y PTC, desarrolló soluciones escalables de realidad mixta que sobreponen datos en tiempo real de productos conectados con experiencias de realidad aumentada 3D, para auxiliar a sus ingenieros en la atención y solución de problemas con algunos equipos (Wilson, 2021). El fabricante de acero Severstal, recurrió a la empresa General Electric (GE) para desarrollar una solución basada en IoT que permitiera reducir las demoras por procesos de mantenimiento no programados (GE Digital, 2021a).

Transporte o movilidad

Uno de los más notables desarrollos en esta área es el de los automóviles autónomos, con una gran competencia no solo entre las, hasta hoy, principales empresas automotrices, sino también con algunas de reciente creación y de menor tamaño. Entre las más avanzadas destaca la empresa Tesla, con la incorporación al mercado de una nueva generación de vehículos autónomos. También en este segmento destacan el desarrollo de soluciones telemáticas y aplicaciones para la administración de flotillas de vehículos, que incluyen sistemas para el diagnóstico y monitoreo de las unidades. Algunos ejemplos en esta área son KWRL Transportation Co-op, una plataforma para rastreo, monitoreo y administración de rutas de autobuses escolares desarrollado por la empresa Samsara (Samsara, 2021); Onnibus.com de la empresa Telia, plataforma diseñada para optimizar el funcionamiento de flotas de autobuses reduciendo el consumo de combustible procesando datos generados en tiempo real (Telia, 2020) y Caledonian Logistics, diseñado por Geotab, sistema especializado en la distribución de mercancía paletizada que incluye funciones para monitoreo y seguimiento del comportamiento de los conductores (Geotab, 2020).

Energía

La creciente demanda de energía a nivel mundial ha llevado también a incursionar en el IoT buscando soluciones más eficientes e inteligentes desde su generación hasta la distribución. Tal es el caso de la empresa Exelon, que mediante el uso de la plataforma Predix de GE logró un aumento del 70% en el rendimiento de sus parques eólicos (GE Digital, 2021b). También la empresa Enel, utilizó la aplicación de mantenimiento predictivo C3.ai en cinco centros de control para mejorar la confiabilidad de la red y reducir la ocurrencia de fallas (C3.ai, 2021). Para su funcionamiento, esta aplicación utiliza inteligencia artificial para analizar datos de sensores de red en tiempo real, datos de medidores inteligentes, registros de mantenimiento de activos y datos meteorológicos.

Venta minorista

Los minoristas están convencidos de que pueden mejorar su eficiencia en costos y la experiencia de los clientes en tiendas mediante el uso de aplicaciones innovadoras basadas en IoT. Las soluciones en esta área incluyen: la señalización digital en tienda, el seguimiento y la participación de los clientes, el control de las mercancías y la administración de inventario, las máquinas expendedoras inteligentes, entre otros. Por ejemplo, la empresa nuMedia Innovations, cuenta con la solución PRSONAS que mediante el uso de hologramas ofrece servicios de anfitriones y porteros virtuales, los cuales actúan de manera inteligente e interactiva como representantes de ventas virtuales y especialistas en productos en los escaparates de tiendas minoristas (Digi, 2021). También, la empresa RetailNext tiene una solución basada en IoT para comparar el rendimiento de todas sus tiendas de manera muy precisa; su solución utiliza sistemas confiables para revisar métricas detalladas, KPI y analizar las tasas de conversión por tienda, identificando fácilmente normas, tendencias y valores atípicos, además de crear planes para capturar oportunidades (RetailNext, 2021).

Ciudades

Ciudades alrededor del mundo también han avanzado en la adopción del IoT. Para tener un mejor conocimiento de este crecimiento, en 2017 el Instituto para la Administración del Desarrollo (IMD por sus siglas en inglés) y la Universidad de Tecnología y Diseño de Singapur crearon un índice de ciudades inteligentes. Este índice estudia aspectos económicos y tecnológicos de las referidas ciudades, a la par de dimensiones humanas como calidad de vida, medio ambiente, inclusión, entre otros. En su tercera edición, correspondiente al año 2021, destacan Singapur, Zurich y Oslo como las ciudades con mayores avances en estas áreas. Los proyectos que destacan en este segmento tienen que ver con aplicaciones para el control y manejo eficiente de tráfico, servicios públicos como iluminación y manejo inteligente de desperdicios, seguridad pública y monitoreo de las condiciones ambientales (IMD, 2021). Por ejemplo, la ciudad de Ámsterdam cuenta con un sistema de iluminación inteligente en la plaza Hoekenrodeplein el cual se controla de manera remota y se adapta de manera automática a diferentes condiciones de iluminación (LUCI, 2021). Otro ejemplo es la plataforma de sensores Nación Inteligente de Singapur; la cual es capaz de recolectar, analizar y compartir información tomada de sensores y dispositivos que sirven para mejorar la planificación urbana, el sistema de transporte y la seguridad pública (GOVTech, 2021).

Oportunidades y retos

El avance hasta hoy en el área del IoT es muy notable, prueba de esto es el gran número de soluciones comerciales e industriales disponibles en el mercado. Sin embargo, aún hay varios retos por resolver y también, conforme se logran avances, nuevas áreas de oportunidad se identifican. Existen una gran cantidad de artículos que reportan resultados del análisis de los retos y oportunidades desde diferentes sectores o perspectivas (Biswas y Giaffreda, 2014; Hassanalieragh et al., 2015; Ramachandran y Krishnamachari, 2018; Javaid et al., 2018). De manera general, algunas de las áreas en las que se pueden clasificar son las siguientes (Colakovic y Hadzialic, 2018):

- *Interoperabilidad e integración.* La falta de estándares resulta en dificultades para mejorar la interoperabilidad e integración entre sistemas que utilizan tecnologías diferentes. De ahí la importancia de continuar trabajando para el desarrollo de estándares que sean accesibles, tanto para el público en general como para los fabricantes, y que cuenten con el respaldo de organizaciones reconocidas como el IEEE, W3C, IETF, entre otros.
- *Disponibilidad y confiabilidad.* En muchos de los casos, los sistemas IoT necesitan estar disponibles en cualquier lugar y en cualquier momento, así como ofrecer un alto grado de confiabilidad respecto de los servicios que soportan, incluyendo ambientes altamente heterogéneos y dinámicos. Por esto, aún es necesario resolver algunos retos que tienen que ver con la disponibilidad y cobertura de las redes de comunicación, principalmente en las móviles e inalámbricas, así como la interacción o acceso a servicios e infraestructura disponibles en la nube.
- *Almacenamiento de datos y procesamiento.* De forma cada vez más frecuente se hace necesario acceder a servicios, generalmente disponibles en la nube, para el almacenamiento y procesamiento de los datos recolectados por los sistemas IoT. Entendiéndose que procesar no tiene solamente que ver con el almacenamiento de los datos, sino también con capacidades avanzadas de procesamiento como aquellas relacionadas con el *Big Data* y la inteligencia artificial, así como la visualización o representación de los datos, que permitan incluso incorporar o incrementar el grado de autonomía de las aplicaciones IoT, reduciendo o hasta eliminando la necesidad de interacción humana.
- *Potencia y consumo de energía.* De manera particular para dispositivos muy pequeños y móviles o sin acceso permanente a una infraestructura, el contar con tecnologías para el aprovisionamiento confiable y durable de energía, así como potencia suficiente para extender el alcance o cobertura de comunicación, aún son retos en los cuales,

aunque hay avances, todavía quedan aspectos por resolver y nuevos que se generan a raíz de los avances en la miniaturización de componentes.

- *Seguridad y privacidad*. Un área muy importante para cualquier sistema digital, y el IoT no es la excepción, aún más por la complejidad de su diseño ante el gran número de tecnologías y sistemas con los que generalmente interactúan y comparten información este tipo de sistemas. Se requieren más y mejores esquemas de autenticación, encriptación, integridad de la información, así como aspectos de tipo social, legal y cultural.

Programas de la Facultad de Telemática que incorporan IoT como asignatura

La formación de profesionales altamente capacitados en el área del IoT es de extrema importancia; por ello, en la Facultad de Telemática se ofrecen las carreras de Ingeniería en Software e Ingeniería en Tecnologías de Internet, además del programa de Maestría en Tecnologías de Internet. En todos estos hay materias relacionadas con el Internet de las cosas, bajo los nombres que se listan en la tabla 1.

Tabla 1. Materias relacionadas con Internet de las cosas, en la Facultad de Telemática

Materia	Programa académico	Semestre en el que se ofrece
Fundamentos de Tecnologías de Internet	Ingeniería en Tecnologías de Internet	Primero
Internet de las cosas y Cómputo cognitivo	Ingeniería en Tecnologías de Internet	Quinto
Sistemas embebidos para Internet de las cosas	Ingeniería en Tecnologías de Internet	Sexto
Dispositivos inteligentes (optativa)	Ingeniería en Tecnologías de Internet	Séptimo
Programación para Internet de las cosas (optativa)	Ingeniería en Tecnologías de Internet	Séptimo
Seguridad en Internet de las cosas (optativa)	Ingeniería en Tecnologías de Internet	Séptimo
Internet de las cosas (optativa)	Ingeniería de Software	Quinto
Big data y *Data mining*	Ingeniería de Software	Sexto
Arquitectura de sistemas digitales para Internet de las cosas	Maestría en Tecnologías de Internet	Primero
Internet de las cosas	Maestría en Tecnologías de Internet	Segundo
Desarrollo de *software* para Internet de las cosas (optativa)	Maestría en Tecnologías de Internet	Tercero o cuarto
Seguridad en Internet de las cosas (optativa)	Maestría en Tecnologías de Internet	Tercero o cuarto
Aplicaciones de Internet de las cosas (optativa)	Maestría en Tecnologías de Internet	Tercero o cuarto

Fuente: Elaboración propia.

Como se aprecia en la tabla 1, algunas materias son obligatorias en el plan curricular respectivo y otras son materias optativas, con la finalidad de que el estudiante elija el perfil o área de su interés. Particularmente, la elección de estos cursos le permitirán tener bases teóricas y prácticas sólidas para resolver necesidades y problemáticas en diversos entornos de aplicación, relacionadas con el Internet de las cosas.

Conclusiones

Sin duda, los avances tecnológicos para IoT han rebasado por mucho la visión que en 1999 Kevin Ashton conceptualizó, a tal grado que ya se hace referencia hacia áreas más específicas como el Internet industrial de las cosas o, incluso, algunas más actualizadas como el Internet de todo. Hoy, ya existe un importante número de aplicaciones y soluciones IoT en el mercado orientadas tanto a la industria como al consumidor general, en áreas tan importantes como la manufactura, el transporte, la energía, la venta minorista y ciudades, entre otras, temas que se abordan en los siguientes capítulos.

Sin embargo, aún existen varios retos pendientes por resolver para desarrollar aplicaciones que sean más seguras, con capacidad para fácilmente interoperar con otras, que sean confiables, que hagan uso más eficiente de la energía y potencia para la transmisión de los datos, así como mejores capacidades para la integración con servicios o infraestructuras virtuales de la nube. Con lo anterior, tanto para el desarrollo de nuevas soluciones tecnológicas, como de propuestas para la solución de los retos aún pendientes, la formación de profesionales con los conocimientos y habilidades para incursionar en las diferentes áreas del IoT es muy importante, de ahí la necesidad de impulsar desde las universidades la generación de estos profesionistas.

Referencias

Ashton, K. (2015). *How to fly a horse: the secret history of creation, invention, and discovery*. Doubleday.

AVSystem. (2021). *What is Internet of things (IoT)? Everything you need to know*. https://www.avsystem.com/blog/ what-is-internet-of-things-explanation/

Biswas, A. y Giaffreda, R. (2014). IoT and cloud convergence: opportunities and challenges. En *2014 IEEE World Forum on Internet of Things (WF-IoT)* (pp. 375-376). doi: 10.1109/WF-IoT.2014.6803194.

Chui, M., Loffer, M. y Roberts, R. (2010). The Internet of things. *McKinsey Quarterly 2*(2), pp. 1-9.

C3.ai. (2021). *Largest production deployment of AI and IoT applications*. https://c3.ai/customers/enel/

Colakovic, A. y Hadzialic, M. (2018). Internet of things (IoT): a review of enabling technologies, challenges, and open research issues, computer networks. doi: 10.1016/j.comnet.2018.07.017

Digi. (2021). *NuMedia unveils next generation of PRSONAS digital hosts – Powered by Digi*. https://www.digi.com/customer-stories/ numedia-interactive-self-service-holograms

Fernández, R. (2022). El Internet de las cosas (IoT) - Datos estadísticos. *Statista*. https://es.statista.com/temas/6976/el-internet- de-las-cosas-iot/#dossierKeyfigures

GE Digital. (2021a). *Severstal reduces unscheduled maintenance delays by 20% with predix asset performance management*. https://www.ge.com/digital/customers/severstal-reduces-unscheduled-maintenance-delays-20-asset-performance-management

GE Digital. (2021b). *Exelon optimizes wind forecasting accuracy*. https://www.ge.com/digital/customers/exelon-optimizes-wind- forecasting-accuracy#to-section-index=section-5

Geotab. (2020). *Caledonian Logistics: logistics company seeks reliable driver monitoring*. https://www.geotab.com/case-study/caledonian- logistics/

GOVTech. (2021). *Smart nation sensor platform*. https://www.tech.gov.sg/products-and-services/smart-nation-sensor-platform/

Hassanalieragh, M., Page, A., Soyata, T., Sharma, G., Aktas, M., Mateos, G., Kantarci, B. y Andrescu, S. (2015). Health monitoring and management using Internet of things (IoT) sensing with cloud-based processing: opportunities and challenges. En *2015 IEEE International Conference on Services Computing* (pp. 285-292). doi: 10.1109/SCC.2015.47.

Javaid, N., Sher, A., Nasir, H. y Guizani, N. (2018). Intelligence in IoT-based 5G networks: opportunities and challenges. *IEEE Communications Magazine*, *56*(10), pp. 94-100. doi: 10.1109/MCOM.2018.1800036.

Langley, D., van Doorn, J., Ng, I., Stieglitz, S., Lazovik, A. y Boonstra, A. (2021). The Internet of everything: smart things and their impact on business models. *Journal of Business Research*, *122*, pp. 853-863. https://doi.org/10.1016/j.jbusres.2019.12.035

Lasse Lueth, K. (2020). *Top 10 IoT applications in 2020. IOT Analytics Market Insights for the Internet of Things*. https://iot-analytics.com/top-10-iot-applications-in-2020/

LUCI Association. (2021). *Smart lighting for Amsterdam's Hoekenrodeplein square*. https://www.luciassociation.org/smart-lighting-for-amsterdams-hoekenrodeplein-square/

Ramachandran, G. y Krishnamachari, B. (2018). *Blockchain for the IoT: opportunities and challenges. Computer science, distributed, parallel, and cluster computing*. Cornell University.

RetailNext. (2021). *Resources*. Consultado el 16/11/2021 de https://retailnext.net/ resources

Samsara. (2021). *Case study KWRL Transportation Co-op. The leader in Industrial IoT*. https://corpweb-static.s3-us-west-2.amazonaws.com/pdf/docs/Samsara-Case-Study-KWRL.pdf

Telia. (2020). Fuel Savings and better driving habits for onnibus. https://business.teliacompany.com/blog/Fuel-savings-and-better- driving-habits-for-OnnibusApellidos

Veneri, G. y Capasso, A. (2018). *Hands-on Industrial Internet of things*. Editorial Packt Publishing.

Wilson, M. (2021). Howden creates mixed reality solutions to enhance customer experience. Leveraged Vuforia Studio to improve service and reduce customer downtime: case studies. *Ptc.com* https://www.ptc.com/en/case-studies/howden-mixed-reality

Capítulo 2

Ecosistemas de IoT

José Román Herrera Morales
Sara Sandoval Carrillo
Armando Román Gallardo

Resumen

Cuando nos referimos al gran crecimiento y desarrollo de aplicaciones de IoT, es frecuente encontrar el término de "ecosistema IoT" mencionado como un conjunto de diferentes elementos o actores que están relacionados para llevar a cabo su función y permitir así el logro de las tareas para las cuales una aplicación de IoT fue diseñada. Sin embargo, este concepto de ecosistema IoT generalmente se da por entendido y no se describen todos los elementos que están presentes, o se aborda desde diferentes perspectivas sin definirlo claramente. Por ello, en este capítulo presentamos diferentes definiciones que son producto de la revisión de literatura, tanto de artículos científicos como diferentes recursos web publicados en portales especializados, mismos que sitúan al ecosistema de IoT como un modelo de negocios desde una óptica empresarial, hasta otras conceptualizaciones con una visión industrial, en la que el ecosistema representa una solución integral basada en diversos componentes con una arquitectura integrada en capas, a manera de analogía entre las plataformas de IoT y los ecosistemas afines en las que las grandes empresas tecnológicas juegan un rol preponderante, ofreciendo su infraestructura y so-

luciones integrales para llevar a cabo un sinfín de aplicaciones y servicios de IoT.

Introducción

Como se ha mencionado, cuando se habla del IoT o se revisa información publicada sobre el desarrollo de servicios y aplicaciones para ello, es muy común que se refieran al término de "ecosistema IoT" (en inglés, IoT *ecosystem*) como un conjunto de diferentes elementos, actores o tecnología que están interrelacionados para generar aplicaciones IoT hacia un dominio en particular. Sin embargo, generalmente no se brinda una definición formal del concepto o no se describe de forma clara bajo qué perspectiva se está analizando este tema. Debido a ello, en este capítulo se realiza un análisis de diferentes definiciones y propuestas arquitectónicas para referirse a los ecosistemas IoT, y se describen desde perspectivas diferentes, a decir: 1) el ecosistema IoT conformado por componentes clave que son vistos como una arquitectura de capas o niveles funcionales, 2) el ecosistema IoT como si fuese un modelo empresarial o de negocios y 3) una analogía entre los ecosistemas IoT vs las plataformas IoT, donde en este último enfoque coinciden diversas empresas como Amazon, Microsoft, etc., las cuales son influyentes en el área con sus respectivas plataformas y servicios comerciales.

Ecosistema de IoT desde una perspectiva empresarial y como un modelo de negocios

Leminen et al. (2018) identifican una fuerte conexión entre los ecosistemas y los tipos de modelos de negocio para aplicaciones de IoT, y consideran que se requiere ampliar la concepción del punto de vista de una sola empresa para llevarlo a una perspectiva más amplia con el enfoque de un ecosistema. La conexión entre los ecosistemas y los tipos de modelos de negocio se ha evidenciado en la literatura reciente, donde Tarkoma y Katasonov (2011) definen que un ecosistema para IoT es un conglomerado de empresas e individuos donde las primeras utilizan un conjunto común de activos, basados en la interconexión de un mundo físico de las "cosas" con un mundo virtual de Internet.

No obstante, ante el surgimiento de ecosistemas emergentes de IoT, existen desafíos relacionados con la diversidad de objetos, la inmadurez de la innovación y los ecosistemas no estructurados (Westerlund et al., 2014) que, junto con la evolución tecnológica y las características específicas de las aplicaciones emergentes para el IoT, provocan que los modelos de negocios también se vayan adaptando a los nuevos servicios que se están generando. Asimismo, de acuerdo con la visión de Mazhelis et al. (2012): El ecosistema empresarial de IoT es un tipo especial de ecosistema que se compone de empresas e individuos relacionados con la IoT, que interactúan junto con su entorno socioeconómico para ayudar a las empresas a expandir sus mercados y estimular la innovación para la generación de nuevos productos y servicios.

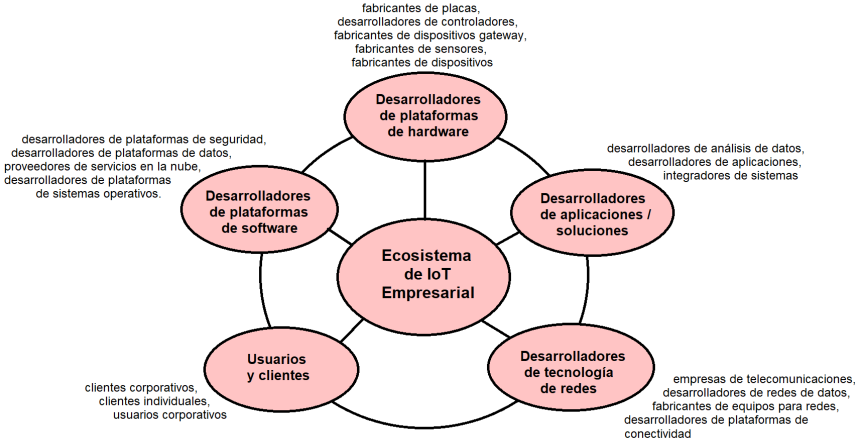
Por tanto, si tomamos como base el ecosistema para IoT desde la perspectiva empresarial, se pueden identificar cinco elementos clave para este tipo de ecosistema (Lee, 2019): los desarrolladores de plataformas de *software*, los desarrolladores de plataformas de *hardware*, los desarrolladores de tecnología e infraestructura de red, los desarrolladores de soluciones y aplicaciones y, por último, están los clientes y usuarios finales de estos servicios y aplicaciones. En la Figura 1 se muestra la interacción entre estos cinco principales actores.

A continuación, se dan detalles de cada uno de estos elementos clave de un ecosistema empresarial para IoT descrito por Lee (2019):

- *Desarrolladores de plataformas de software*. En este segmento se incluye a las plataformas especializadas en gestión de datos y en seguridad, a los proveedores de servicios basados en la nube e incluso a las plataformas de sistema operativo para los dispositivos IoT. Las empresas más representativas en este rubro son Microsoft, Amazon, IBM, Google, entre muchas otras. La utilización de estas plataformas integradas de *software* representa una gran ventaja competitiva para los desarrolladores de aplicaciones de IoT, puesto que reduce la complejidad y agiliza el desarrollo, la implementación y la gestión de aplicaciones durante todo el ciclo de desarrollo de

software, debido a que simplifican tareas comunes como la limpieza, transformación, almacenamiento e integración de datos, también facilitan la disponibilidad y escalabilidad de las aplicaciones, puesto que se integran con servicios de cómputo en la nube que utilizan bajo demanda recursos virtuales para incrementar la capacidad de almacenamiento y procesamiento de información.

Figura 1. Actores clave de un ecosistema empresarial para IoT



Fuente: Elaboración propia, adaptado de Lee (2019).

Además, estas plataformas de *software* proporcionan *frameworks* para programación de aplicaciones que emplean APIs estandarizadas como una capa de abstracción, que facilitan la portabilidad entre diferentes plataformas de *hardware* y sistemas operativos a fin de interconectar de forma transparente a sensores, actuadores y dispositivos, en los que se aplican mecanismos de seguridad y traducción de protocolos de comunicación de datos que garantizan la interoperabilidad de las aplicaciones.

- *Desarrolladores de plataformas de hardware*. El crecimiento de IoT se ha visto favorecido por la oferta de una amplia gama de dispositivos de *hardware* que cada vez son de menor tamaño, de bajo costo, con mayor autonomía energética e, incluso, algunos de estos componentes cuentan con microcontroladores que le añaden capacidades de preprocesamiento de datos, lo cual facilita el sensado y adquisición de datos para las aplicaciones IoT. Destacan en este mercado de plataformas de *hardware* (Singh y Kapoor, 2017) las tarjetas o placas integradas, los sensores, los actuadores y demás dispositivos periféricos de fabricantes reconocidos como: Raspberry Pi, Espressif, Adafruit, Particle, BeagleBone, Arduino, IBM, Intel Edison, Samsung Artik, entre otros, quienes además ofrecen kits y plataformas con entorno de desarrollo (IDE) para facilitar la integración de sus dispositivos físicos con el código de programación necesario dirigido a agilizar el desarrollo de las aplicaciones IoT.
- *Desarrolladores de tecnologías de redes*. Las tecnologías de telecomunicaciones que permiten la interconexión de red para las aplicaciones IoT son proporcionadas en cada país por las principales empresas de telecomunicaciones, las cuales son desarrolladores de plataformas de conectividad, de redes de datos y fabricantes de los dispositivos e infraestructura de red. Estos proveedores de conectividad buscan estar a la vanguardia, ofreciendo las tecnologías de telecomunicaciones que mejor se apliquen para cada escenario de aplicación para IoT. Las tecnologías de comunicación empleadas en IoT son muy diversas y están en constante desarrollo, se pueden dividir de acuerdo con la distancia de alcance entre sus dispositivos; pueden ser de corto o largo alcance, en las que las distancias van desde algunos metros hasta varios kilómetros. Para comunicaciones que son de corto alcance (Bello et al., 2017), conocidas como de dispositivo a dispositivo (D2D) o de máquina a máquina (M2M), no se requiere tener una infraestructura de red intermedia,

sino que la conexión es directa entre cada dispositivo. Las plataformas de comunicación D2D más populares son: Zigbee, RFID (identificación por radiofrecuencia), Bluetooth, BLE (Bluetooth Low Energy) y NFC (comunicación de campo cercano). Mientras que para la interconexión de dispositivos de IoT de largo alcance (Sinha et al., 2017) se utilizan tecnologías como: LPWA (tecnologías de área amplia de baja potencia), NB-IoT (NarrowBand-Internet of things), LTE-M (Long Term Evolution for Machines), Sigfox, LoRa y, recientemente, la tecnología de banda ancha móvil 5G. Las cuales no necesariamente son del mismo estándar, pero su aplicación es semejante.

- *Desarrolladores de aplicaciones y soluciones*. El proceso de desarrollo de aplicaciones para IoT no es sencillo y requiere de profesionales e ingenieros con mayor experiencia, dado que involucra integrar diversas plataformas, dispositivos y tecnologías que operan en un ambiente distribuido. Para ello, estos integradores de sistemas y desarrolladores de soluciones evalúan y utilizan las plataformas de IoT que mejor se amolden a la necesidad específica de la solución a desarrollar, donde es común que se combinen plataformas de *software* con dispositivos físicos de diferentes fabricantes, por ejemplo: pueden tenerse placas de Arduino o Raspberry Pi interconectados y monitorizados desde la plataforma de AWS IoT o de Microsoft Azure IoT. También, este rol de especialistas e integradores de soluciones para IoT va tomando cada vez más relevancia; según reporte de la firma ABI Research (2017), se estima que los ingresos de consultoría e integración de sistemas IoT tendrán una tasa de crecimiento anual del 16.1%, esto considerando los resultados analizados en el periodo de 2017 a 2022, por lo que es de esperar que profesionales y empresas especializadas en ofrecer soluciones de extremo a extremo serán cada vez más solicitadas en el mercado del IoT.

- *Usuarios y clientes*. Los usuarios de aplicaciones y servicios de IoT pueden ser desde usuarios individuales, como cualquier persona que tenga en su casa algún dispositivo IoT para tareas simples como prender las luces o activar o desactivar algún electrodoméstico, hasta usuarios o clientes empresariales o corporativos, que utilizan dispositivos y aplicaciones IoT para el monitoreo o automatización de procesos comerciales, industriales y de servicios.

Ecosistema IoT como un modelo de capas y componentes

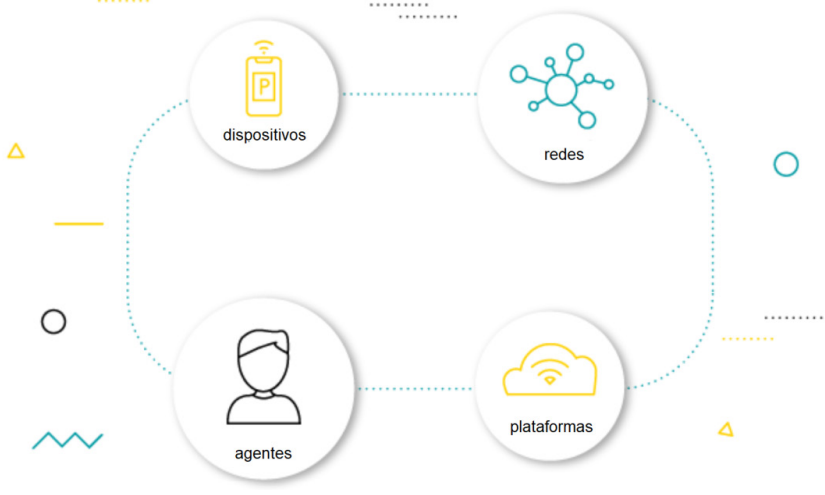
Si consultamos referencias con recursos no tan formales, como pueden ser artículos publicados en *blogs* web o información técnica publicada por empresas especializadas en el tema, también ellos ofrecen su perspectiva y definiciones sobre el concepto de ecosistema IoT. En algunos casos, como Memmet Yildhiz (2020) definen a un típico ecosistema IoT como una solución integral conformada por sistemas simples o complejos que se integran por componentes y subcomponentes, para llevar a cabo simples o múltiples tareas, pero trabajando de forma coherente y en armonía. En esta perspectiva de Memmet, el concepto de ecosistema lleva cierta analogía a los mecanismos metafóricos de supervivencia y requisitos de integración similares a los que están presentes en los ecosistemas biológicos.

En otras definiciones, como la propuesta por la empresa Qmansys (Pal Bijoy, 2020) se describe al ecosistema de IoT como un sinónimo de arquitectura, donde los componentes de este serían los ladrillos que sirven para la construcción de esa edificación. También, en el portal de Thalesgroup.com (2021) se describe a un ecosistema IoT desde una visión industrial integrada por capas clave de componentes y, en esta misma línea, la empresa AVSystems (2021) lo refiere como una integración de diversos componentes clave, integrados en una arquitectura de elementos definidos como una extensa red de dispositivos, tecnologías y plataformas inter-

conectados e interdependientes que han sido integrados para lograr un objetivo muy particular; es decir, se podrían tener diversos ecosistemas IoT según las diferentes aplicaciones de IoT que se implementen, entre las que se podrían mencionar: ciudades inteligentes, granjas automatizadas, sistemas de riego inteligente, rastreo y logística de servicios de transportes, monitorización de procesos de fabricación, entre muchas otras.

Tomando esta última definición de Avsystem.com (2021), en una esquematización más simple, en un ecosistema IoT se tienen dispositivos que recopilan datos y los envía a través de la red a una plataforma que agrega los datos para uso futuro por parte de las personas. Lo que nos lleva a la integración en un ecosistema IoT basado en cuatro componentes fundamentales: dispositivos, redes, plataformas y agentes, que se muestran en la figura 2 y se describen con detalle a continuación.

Figura 2. Ecosistema IoT de 4 componentes



Fuente: Elaboración propia basado en AVsystem.com (2021).

- *Dispositivos*. Una aplicación requiere de diferentes dispositivos IoT, mismos que son clasificados de forma general entre sensores y actuadores. Los sensores son

dispositivos de *hardware* que se emplean para detectar cosas, como temperatura, movimiento, partículas, etc., y los dispositivos actuadores, como su nombre lo indica, accionan o actúan sobre las cosas, como el caso de interruptores, motores, entre otros. En la práctica, los dispositivos IoT puede que no sean simples sensores o lectores de una variable física como la temperatura o humedad, y los actuadores no ser simples interruptores digitales, sino que pueden llegan a estar conformados por un conjunto de componentes electromecánicos para llevar a cabo tareas muy sofisticadas como telemedicina, telemonitoreo, procesos de sensado y control muy sofisticados que combinan una gran cantidad de sensores y actuadores que han sido diseñados para trabajar juntos. Adicionalmente, los sensores y actuadores requieren de dispositivos de interfaz o de interconexión hacia los sistemas de redes o telecomunicaciones, a estos dispositivos se les conoce como pasarelas, *gateways* o *drivers*, y representan precisamente el enlace con el siguiente componente del ecosistema de IoT.

- *Redes*. La conectividad entre los dispositivos IoT se da gracias a diferentes tecnologías de red, que van desde las populares redes inalámbricas Wi-Fi y Bluetooth, hasta tecnologías más especializadas, de corto alcance como NFC (*Near Field Communication*), Z-wave, ZigBee y otras redes de cobertura más extensa área, como 5G, LTE, LoRaWAN, SigFox y LPWAN (*Low Power*-WAN). Actualmente, existe una gran variedad de tecnologías a elegir, que difieren en varios aspectos, como el rango de cobertura, la velocidad de la transferencia de datos, el consumo y correspondiente duración de la fuente de energía, etc., por lo que la adecuada selección deberá estar acorde a las necesidades de implementación IoT que se vaya a realizar.
- *Plataformas*. Las plataformas de IoT son siempre el factor aglutinante de cualquier ecosistema de IoT, ya sea que estén en la nube o no. Mediante estos componen-

tes tecnológicos, se realiza la gestión del ciclo de vida de los dispositivos, la recopilación, extracción y procesamiento de los datos, a fin de darles sentido y poder interpretarlos para facilitar la toma de decisiones. Afortunadamente, cada vez se cuenta con una mayor oferta en el mercado de plataformas integrales para IoT, por lo que la selección de la plataforma IoT "ideal" no es tarea fácil y no debe tomarse a la ligera, puesto que podría determinar el éxito o fracaso de una implementación. La plataforma IoT debe tener ciertas características, entre las cuales se establece: versatilidad, adaptabilidad a cambios, y que puede ser escalable, para que el ecosistema de IoT pueda crecer de forma natural conforme las necesidades de implementación de la aplicación vayan madurando también.

- *Agentes*. Los agentes son todas las personas cuyas acciones afectan el ecosistema IoT. Estos agentes van desde los ingenieros que diseñan y materializan las plataformas de IoT, hasta los operadores o usuarios finales de las mismas. Pero también, parte de los actores y agentes más importantes son los clientes, quienes buscan resolver una necesidad o problemática. Sin duda, el componente tecnológico es importante: los sensores, las redes, las plataformas, pero toda esta integración como un ecosistema no tendría mucho valor si no fuera precisamente por las personas que los crean, en busca de impulsar la eficiencia y mejorar la calidad de vida y de procesos existentes.

Finalmente, se tiene un tercer enfoque para describir el concepto de ecosistema IoT y este es el de relacionarlo directamente a la oferta comercial que ofrecen las plataformas especializadas para IoT de las grandes empresas tecnológicas como Amazon, Microsoft, Google, entre otras. Especialistas del área como Bruce Sinclair (2016) hacen un análisis de los ecosistemas de IoT y los comparan, asociando este concepto a la oferta que realizan

las empresas fabricantes y desarrolladoras de tecnología, mencionando que, mientras las plataformas de IoT están en el ámbito de lo técnico, el ecosistema de IoT está en el ámbito de los negocios. El ecosistema de IoT es visto como una comunidad de proveedores y empresas vinculadas entre sí, para monetizar los productos de IoT y ofrecer soluciones más cercanas a las necesidades del cliente. Compartiendo esta visión de ecosistema IoT como plataforma tecnológica, Microsoft Azure IoT (Microsoft, 2022), emplea este término como un ecosistema de miles de asociados (*partners*), los cuales son empresas proveedoras de servicios especializados de IoT, lo que le permite a sus clientes elegir una solución totalmente administrada o hecha a la medida, así como los conocimientos y recursos que faciliten la estimación del costo de nuevas soluciones que mejor se amolden a las necesidades de su negocio.

Conclusiones

Como se describe a lo largo de este capítulo, para favorecer el desarrollo del IoT y sus aplicaciones en diferentes sectores como la industria, la salud, el gobierno y demás sectores económicos, debe abordarse desde una perspectiva integral como un ecosistema. El concepto de ecosistema para IoT puede verse desde diferentes maneras, desde un enfoque basado en modelo de negocios o empresarial, como una perspectiva de infraestructura y de las grandes plataformas tecnológicas y, también, desde una perspectiva más general en la que se trata como una red compleja de agentes y relaciones interdependientes entre diferentes actores como las personas, capas de tecnología y la propia industria.

El IoT está en franco crecimiento y representa una gran área de oportunidad para las empresas de tecnología en todo el mundo, pero se requiere de más investigación, consenso entre estos grandes actores que tienen el carácter de ser empresas preponderantes, para favorecer el desarrollo y estandarización de plataformas y herramientas de IoT que propicien la innovación, integración, interoperabilidad, y la disponibilidad de servicios y aplicaciones en el mundo real.

A manera de conclusión, se puede considerar que un ecosistema de IoT se trata de ese conjunto de actores y elementos

que están presentes para que, tanto en la industria y en el ámbito de los negocios, se facilite el desarrollo y gestión de aplicaciones de IoT que permitan un mejor funcionamiento y posibilidades de implementación exitosa.

Referencias

ABI Research. (2017). *System integrators quickly becoming the IoT gatekeepers*. Consultado el 7 de enero de 2022 de https://www.abiresearch.com/press/system-integrators-quickly-becoming-iot-gatekeeper/

AVSystem.com. (2021). *IoT ecosystem: 4 key elements*. https://www.avsystem.com/blog/iot-ecosystem/

Bello, O., Zeadally, S. y Badra, M. (2017). Network layer inter-operation of Device-to-Device communication technologies in Internet of Things (IoT). *Ad Hoc Networks*, 57. pp 52-62. https://doi.org/10.1016/j.adhoc.2016.06.010.

Lee, I. (2019). The Internet of things for enterprises: an ecosystem, architecture, and IoT service business model. *Internet of Things*, 7. https://doi.org/10.1016/j.iot.2019.100078.

Leminen, S., Rajahonka, M., Westerlund, M. y Wendelin, R. (2018). The future of the Internet of things: towards heterarchical ecosystems and service business models. *Journal of Business & Industrial Marketing, 33*(6). https://doi.org/10.1108/jbim-10-2015-0206

Mazhelis, O., Luoma, E. y Warma, H. (2012). *Defining an Internet-of-Things Ecosystem*. En S. Andreev, S. Balandin e Y. Koucheryavy (Eds.), *Internet of things, smart spaces, and next generation networking. ruSMART 2012, NEW2AN 2012. Lecture Notes in Computer Science. Vol. 7469* (pp. 1-14). Springer. https://doi.org/10.1007/978-3-642-32686-8_1

Microsoft. (2022). *Azure IoT. La plataforma de Internet de las cosas de Microsoft*. https://azure.microsoft.com/es-mx/overview/iot/

Bijoy, P. (2020). IoT Architecture – Bricks of IoT Ecosystem. *Qmansys Infosolutions*. https://www.qmansys.com/iot-architecture-bricks-of-iot-ecosystem/

Sinclair, B. (2016). IoT ecosystems versus IoT platforms. *IoT-Inc.com* https://www.iot-inc.com/iot-ecosystems-versus-iot-platforms-article/

Singh, K. J. y Kapoor, D.S. (2017). Create your own Internet of things: a survey of IoT platforms. *IEEE Consumer Electronics Magazine*, *6*(2), pp. 57-68. https://doi.org/10.1109/MCE.2016.2640718

Sinha, RS., Wei, Y. y Hwang, S. H. (2017). A survey on LPWA technology: LoRa and NB-IoT. *ICT Express*, *3*(1), pp. 14-21. https://doi.org/10.1016/j.icte.2017.03.004

Tarkoma, S. y Katasonov, A. (2011). *Internet of things strategic research agenda (IoT-SRA).* Finnish Strategic Centre for Science, Technology, and Innovation: for Information and Communications (ICT) Services, Businesses, and Technologies. http://www.internetofthings.fi/extras/internet-of-things-strategic-research-agenda.pdf

Thalesgroup.com (2021). *The IoT ecosystem in 2021: components, industry alliances and legal environments.* https://www.thalesgroup.com/en/markets/digital-identity-and-security/iot/inspired/iot-building-blocks

Westerlund, M., Leminen, S. y Rajahonka, M. (2014). Designing business models for the Internet of things. *Technology Innovation Management Review, 4*(7) pp. 5-14. https://doi.org/10.22215/timreview/807

Yildhiz, M. (2020). Introduction to IoT Ecosystem. A technical, architectural and solution design overview. *Portal Medium*. https://medium.com/technology-hits/introduction-to-iot-ecosystem-25b359c8cf23

Capítulo 3

Componentes principales de *hardware* para IoT

Armando Román Gallardo
Carlos Alberto Flores Cortés
José Román Herrera Morales
Omar Álvarez Cárdenas

Resumen

En este capítulo se describen de forma general los principales componentes de *hardware* que hacen posible el desarrollo de proyectos y aplicaciones para IoT. Estos componentes son los microprocesadores y, de forma más particular, los microcontroladores. Dichos componentes electrónicos representan el núcleo o el cerebro de la aplicación, quienes proporcionan la potencia y capacidad de procesamiento de datos, los cuales tienen que integrarse con otros componentes y circuitos accesorios para leer la información, convertirla, procesarla y poder enviarla a otros dispositivos que forman parte del sistema, empleando un cierto protocolo o tecnología de telecomunicaciones, como pudiera ser Wi-Fi, Ethernet o Bluetooth. Por tanto, en esta revisión, se describen las principales diferencias entre los microprocesadores y microcontroladores, también se menciona la presentación en que estos componentes de *hardware* están disponibles para los usuarios en la amplia oferta de placas de desarrollo que hacen los fabricantes y se describen también elementos como los sensores, transductores, actuadores

y controladores que permiten que en su conjunto podamos contar con la infraestructura de *hardware* electrónico indispensable para proyectos de desarrollo para IoT.

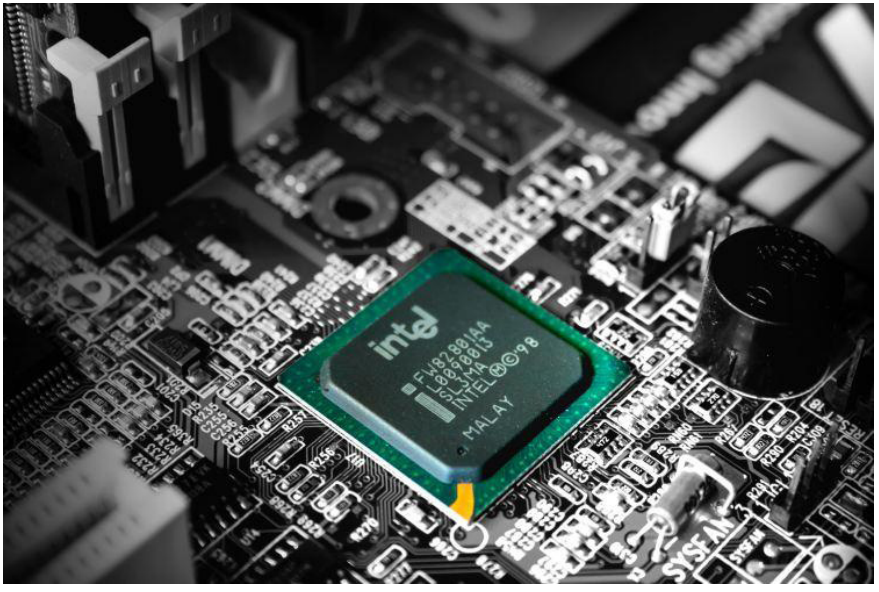
Microprocesadores y microcontroladores

Los componentes más importantes de las computadoras y de muchos dispositivos electrónicos que tienen capacidad de procesamiento de información son los microprocesadores o, de forma más especializada, los microcontroladores. Ambos sirven como el corazón o el núcleo, pero tienen ciertas diferencias que los distinguen entre sí y que se aprecian desde su diseño, tamaño, capacidades y, por supuesto, la gama de aplicaciones para los que están destinados. Los microprocesadores son más potentes, pero deben emplearse como componentes individuales en sistemas más grandes para funcionar, en tanto que los microcontroladores tienen menor potencia de cómputo y se usan para aplicaciones y tareas muy específicas. A continuación, se describe cada uno de estos, para conocer cómo se integran con otros componentes electrónicos en tarjetas o placas de desarrollo y la oferta existente de distintos modelos, presentaciones y fabricantes, a fin de seleccionar los más idóneos de acuerdo a proyectos de desarrollo para IoT.

Microprocesadores

Se llama así a los procesadores de dimensiones muy pequeñas en los que todos los elementos están agrupados en un solo circuito integrado, en este sentido los microprocesadores son componentes electrónicos, los cuales son considerados el cerebro de las computadoras. Algunas de sus tareas principales son las de interpretar las instrucciones que ordena el usuario, la realización de cálculos necesarios y toma de decisiones a través del sistema operativo, todo esto en un lenguaje propio que está en código binario. Un ejemplo de ello es la Unidad Central de Procesos (CPU, del inglés Central Processing Unit) que tiene una arquitectura diseñada para llevar a cabo la ejecución de comandos y secuencias de programas mediante un programa de control (María y Daniel, 2018). En la figura 1 se presenta un ejemplo de microprocesador.

Figura 1. Imagen de un microprocesador Intel



Fuente: Recuperado de https://iotconsulting.tech/como-elegir-un-microcontrolador-para-iot/

Algunas características de los microprocesadores son:

- Se encuentran recubiertos por una cerámica que protege al silicio de algunos elementos como el oxígeno que se encuentra en el aire.
- Cuentan con un caché, que es una memoria ultrarrápida para atender de una manera adecuada y eficiente la gran cantidad de datos que recibe para procesarse.
- Cuentan también con un coprocesador matemático el cual se encarga de realizar las operaciones lógicas, formales y los cálculos con valores de punto flotante.
- Para llevar el control de su propio funcionamiento y controles, tienen una pequeña memoria a la cual se le da el nombre de registro.
- Para comunicar la información con los demás dispositivos, cuentan con una serie de canales de comunicación que comúnmente se denominan puertos (Bengochea Guevara et al., 2020).

Los microprocesadores tienen capacidades mayores que los microcontroladores, por lo que ejecutan más operaciones por segundo, lo que significa que realizan una cantidad mayor de operaciones por unidad de tiempo; por consiguiente, son de una mayor capacidad y velocidad con respecto a las acciones realizadas por un microcontrolador y, debido a esto, su consumo de energía es mucho más elevado, lo que representa un gran desafío para los fabricantes al tener que diseñar cada vez microprocesadores que ahorren más energía. En ese sentido, el bajo consumo de energía representa una gran ventaja en los microcontroladores (Ábrego Preza et al., 2018).

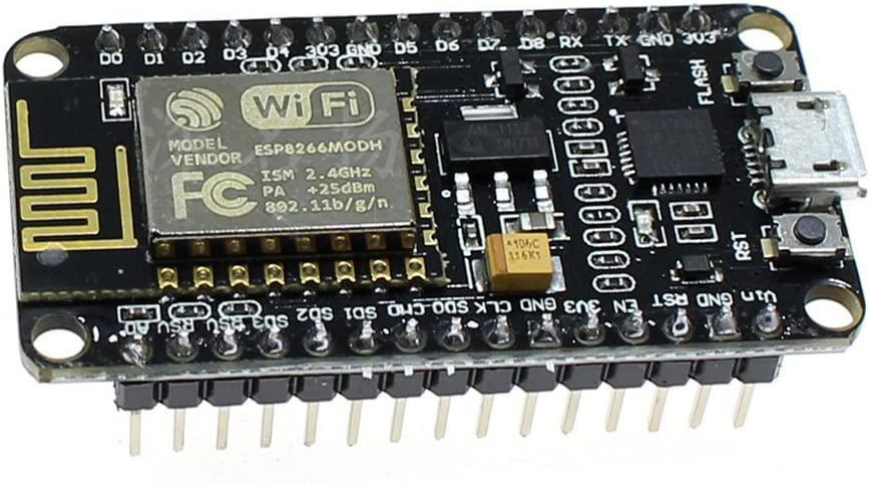
Microcontroladores

Son pequeños dispositivos electrónicos, los cuales pueden configurarse para realizar acciones mediante la programación de instrucciones que realiza el usuario. Tienen gran capacidad de procesamiento, son de bajo costo y una de sus principales ventajas es su menor tamaño, si se compara contra los microprocesadores, lo que permite ahorrar espacio físico al momento de realizar el diseño de proyectos de IoT, mejorando la fiabilidad de funcionamiento y un bajo consumo de energía de los circuitos. Estos microcontroladores están diseñados para emplearse en aplicaciones que interactúan con el medio ambiente, a través del monitoreo de variables que realizan los sensores y tareas de control que pueden llevarse a cabo con los actuadores. Aunque tienen una capacidad menor que los microprocesadores, conforme pasa el tiempo esta ha ido aumentando, a la par de su capacidad de memoria y almacenamiento (Valdés y Areny, 2007).

Para tener una conexión directa con su entorno y otros dispositivos electrónicos, los microcontroladores se comunican directamente con sus periféricos mediante los conversores analógicos a digitales (ADC) y digitales a analógicos (DAC), el bus serial, además de pines digitales y analógicos de propósito general (GPIO). Esto hace posible que necesiten menos capacidad de *hardware* y *software* al realizar acciones como la de leer los valores de los sensores, activar o desactivar una salida, así como para el envío

de mensajes por protocolos como el Message Queing Telemetry Transport (MQTT) o ahora MQ Telemetry Transport (Hunkeler et al., 2008) o uso de microservicios que se encuentran en la nube mediante Wi-Fi. La Figura 2 muestra el microcontrolador en una placa de desarrollo IoT, también llamado NodeMCU.

Figura 2. Placa de desarrollo IoT que utiliza el microcontrolador ESP8266



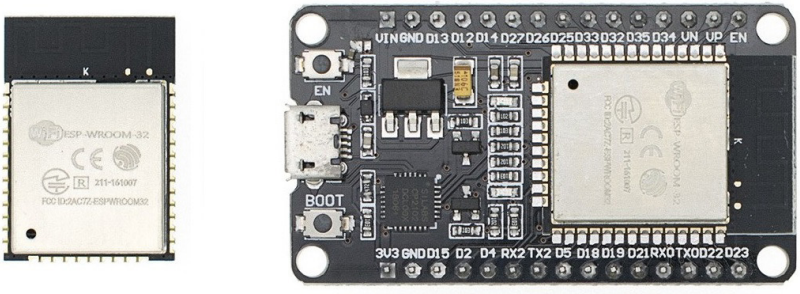
Fuente: Recuperada de https://m.media-amazon.com/images/I/61y-6WNNwBL._AC_SL1001_.jpg

Algunas de sus características tienen que ver con los *bits* que nos indican que el tamaño de las instrucciones puede ser de 8, 16 o 32 *bits*, luego está la referente al tamaño de la memoria que es volátil o no volátil. La memoria *flash* es de tipo no volátil, es decir, los datos que se almacenan en *flash* son persistentes y no se perderán si se apaga el dispositivo IoT y luego se prende de nuevo, aquí es donde se guardan tanto el programa que ejecuta el microprocesador como algunas opciones básicas de configuración inicial de la aplicación. Por el contrario, la memoria volátil, comúnmente mencionada como RAM se emplea para almacenar temporalmente datos, pero, si se interrumpe la energía eléctrica o se reinicia el dispositivo, todos los datos que estaban en ella se perderán. La

cantidad de memoria es un factor que incide en el precio de los dispositivos, entre más memoria tenga un microcontrolador más caro resulta.

Es importante diferenciar también entre lo que son los *chips* o encapsulados de los microcontroladores (MCU, del inglés *Micro-Controller Unit*) y las placas de desarrollo. Las placas de desarrollo incluyen a los microcontroladores junto con circuitería y componentes adicionales, como reguladores de voltaje para protección de la alimentación de entrada, así como un conjunto de interconexiones y puertos de entradas y salidas digitales que tienen conversores analógicos/digitales (ADC - *Analogic-Digital Conversor*) y digitales/analógicos (DAC - *Digital-Analogic Conversor*) que son los pines de las entradas y salidas de propósito general GPIO (por sus siglas en inglés *General Purpose Input Output*). Ver la Figura 3 como referencia a las diferencias entre un módulo de microcontrolador y una placa de desarrollo.

Figura 3. Diferencias físicas entre un módulo de microcontrolador y una placa de desarrollo



Nota: En la izquierda se muestra el módulo ESP32 y a la derecha la placa de desarrollo NodeMCU ESP-32 de 30 pines que a su vez contiene el ESP32.
Fuente: Imagen adaptada de https://www.robotop.lv/en/platformy/626-248-nodemcu-esp-32-v11.html

Otro aspecto importante son las interfaces de comunicación para poder conectarse con otros dispositivos o periféricos, como la interfaz serial I2C (I²C, del inglés *Inter-Integrated Circuit*), 1-Wire (protocolo de comunicaciones en serie diseñado por Dallas Semi-

conductor), SPI (interfaz periférica serial para simplificar las conexiones entre distintos dispositivos), interfaz Ethernet, interfaz Wi-Fi y Bluetooth, entre otras. En la actualidad hay una amplia oferta de placas de desarrollo para IoT y son varios los factores a considerar para decidir qué tipo de placa o tecnología utilizar en cada proyecto de IoT (IoTConsulting.tech, 2019; Manzano, 2021), algunas recomendaciones son: hacer un análisis del consumo y gestión de la fuente de energía, así como del alcance o cobertura de telecomunicaciones, ¿qué tanta distancia debe cubrirse para sensar o transferir datos?, ¿cómo se va a alimentar el dispositivo?, ¿se utilizarían baterías o se tendría una fuente de alimentación continua?, si fuera con baterías, ¿cuánto duraría en funcionamiento?, ¿cuál es su nivel de autonomía energética?, ¿se utilizará este dispositivo en lugares remotos o poco accesibles?

Dispositivos inteligentes para IoT

El incremento exponencial en las capacidades de cómputo de los dispositivos que hoy en día se utilizan en la frontera de los sistemas IoT, es posible incorporar elementos de *software* y *hardware* que agregan "inteligencia" a estos dispositivos. Esto hace posible que los nodos que se instalan en el extremo final de un sistema IoT pasen de ser simples recolectores de información o ejecutores de rutinas, a sistemas con capacidades avanzadas para la toma decisiones e interacción con usuarios u otros dispositivos. Por ejemplo, refrigeradores equipados con cámaras de video en el interior que, combinado con el uso de algoritmos de aprendizaje profundo y procesamiento natural del lenguaje, son capaces de generar o proponer recetas de cocina tomando en cuenta los productos dentro del refrigerador.

Algunos de los principales componentes y sensores que se incorporan en este tipo de sistemas incluyen:

- Componentes de *software* que necesitarán actualizarse.
- Conectividad Wi-Fi.
- Micrófono para recibir comandos de voz.
- *Chip* NFC (*Near Field Communication*) para detectar tarjetas o teléfonos cercanos.

- Conectividad Bluetooth.
- Sensores para detectar humedad, velocidad, movimiento, métricas de salud, entre otros datos.
- Bocinas para mejorar la experiencia de interacción con el usuario.

Con la tecnología actual y disponible en mercado, es posible y fácil la instalación y configuración de un sistema con sensores y cámaras para detectar movimiento en el interior y exterior de una casa, con la capacidad de distinguir entre integrantes de la familia y desconocidos, así como enviar notificaciones de manera automática a los interesados. Si lo trasladamos al área de la salud o del bienestar, mediante el uso de monitores de presión sanguínea, peso corporal, calidad de sueño, calidad del aire, etc., un médico puede delegar el monitoreo de un paciente y recibir notificaciones/alarmas cuando se cumplan condiciones específicas.

El desarrollo de este tipo de sistemas no es exclusivo de los grandes fabricantes, sino que está al alcance de todos, pudiendo implementarse utilizando componentes ya listos para utilizarse como aquellos disponibles en el mercado, hasta dispositivos como los que en este capítulo se presentan, los cuales pueden construirse con algunos conocimientos de programación y electrónica. Este capítulo se centra en el análisis de los componentes electrónicos disponibles para el desarrollo de sistemas IoT, pero también es importante mencionar que el soporte en herramientas y lenguajes de programación que existen es también muy amplio, y va desde librerías de C++ y/o Phyton con capacidades de inteligencia artificial hasta el uso de servicios en la nube para procesamiento avanzado de datos.

Principales plataformas y familias de *hardware* para desarrollo de IoT

En esta sección se realiza un análisis de las principales plataformas que son utilizadas a nivel de experimentación para el desarrollo y prueba de nuevos proyectos y/o ideas de sistemas IoT. Las plataformas electrónicas de desarrollo que han sido seleccionadas son:

Arduino, Raspberry y Espressif Systems. Los criterios de selección son diversos, entre los que destacan: la popularidad y aceptación tanto en el sector educativo como en centros de investigación, así como por aficionados (*hobbistas*); la accesibilidad en términos de su costo y disponibilidad para su adquisición; las capacidades técnicas y el amplio soporte de componentes tanto a nivel de *hardware* como de *software*; y el contar con una gran comunidad de desarrolladores que las utilizan actualmente.

Arduino

Con una propuesta centrada, proveer de *hardware* y *software* de fácil uso para el prototipado rápido, Arduino es hoy en día una de las plataformas de código abierto más populares y reconocidas en el mercado, caracterizada por ser de bajo costo, multiplataforma, contar con un entorno propio para su programación y ser de código abierto con posibilidad de extenderse tanto el *software* como el *hardware*. Debido a esto, se utiliza para la enseñanza en proyectos a nivel primaria o secundaria, pasando por personas aficionadas hasta proyectos científicos complejos. El estudio sobre el impacto y usos de esta plataforma en ambientes educativos a diferentes niveles y en diferentes países está ampliamente documentado, como lo publicado por Lee (2020), El-Abd (2017) y Hurtuk et al. (2017) entre muchos otros.

El catálogo de productos Arduino está organizado en tres grupos o familias[1]:

- *Nano*. Agrupa a las plataformas de desarrollo más compactas en cuanto a su tamaño, pero con amplias opciones para conectividad inalámbrica como Bluetooth y Wi-Fi, así como puertos digitales y analógicos para diferentes propósitos.
- *MKR*. Placas de desarrollo que pueden combinarse para el desarrollo de diferentes tipos de proyectos sin necesidad de agregar circuitería adicional. Además, integrar tecnologías de comunicación más adecuadas para

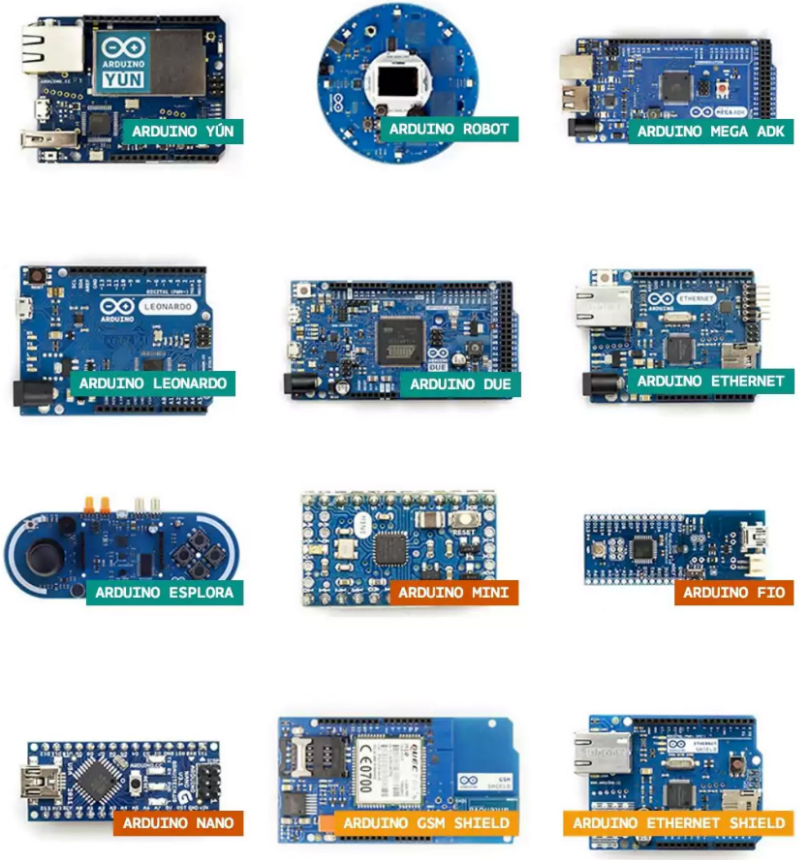
[1] Podrá encontrar una interesante comparativa de estas familias de productos Arduino en: https://en.vmaker.tw/archives/20857

proyectos IoT como Lora y Sigfox, y procesadores de bajo consumo de energía.

- *Classic.* Comprende, como su nombre lo indica, las placas más icónicas y reconocidas de esta marca como Arduino UNO, Leonardo y Micro, consideradas como la base de cualquier proyecto.

En la figura 4 se presentan ejemplos de diferentes placas de Arduino:

Figura 4. Diferentes placas de productos Arduino



Fuente: Recuperada de https://www.circuitschools.com/what-is-arduino-how-it-works-and-what-you-can-do-with-arduino/

Raspberry

Con capacidades muy por encima de las de cualquier placa de desarrollo Arduino, y más cercanas a las computadoras de escritorio modernas, las Raspberry Pi, son también placas de desarrollo ampliamente aceptadas y utilizadas en los sectores educativos, de aficionados e investigadores. Desde su origen, el objetivo de estas plataformas de desarrollo ha sido el brindar acceso a cualquier persona para utilizar computadoras en sus proyectos mediante su oferta de computadoras compactas de muy bajo costo. Su catálogo de productos se divide en dos grandes áreas: hogar e industria, e incluye una amplia gama de dispositivos de *hardware* como computadoras, microcontroladores, cámaras, pantallas y periféricos.

Al igual que Arduino, también cuenta con su *software* propio, en este caso diferentes versiones de un sistema operativo para el funcionamiento de sus computadoras. Uno de sus productos más recientes, el Raspberry Pi 400, se oferta incrustado en un teclado con diversos puertos para conexión de dispositivos periféricos USB y HDMI, para conexión gigabit Ethernet, para memoria microSD, puertos de entrada y salida de propósito general, un microprocesador Broadcom Quad-core de 64 bits a 1.8Ghz con 4 Gb de memoria a un costo de 100 dólares. Con características similares pero un empaque diferente, está el clásico Raspberry Pi 4 Modelo B, que tiene un costo que inicia en los 35 dólares, es el modelo más utilizado y conocido. Para proyectos que requieren un tamaño muy compacto están el Raspberry Pi Zero y el Raspberry Pi Pico, con un costo que inicia en tan solo 5 dólares. El uso de estas plataformas en entornos educativos también está ampliamente documentado en artículos de investigación, tales como los publicados por S. Mahmood et al. (2019), N. S. Yamanoor et al. (2017) y B. Balon et al. (2019), entre otros.

Espressif Systems

Es otro de los fabricantes de módulos o plataformas de desarrollo con características muy similares en cuanto a popularidad, aceptación uso y costos que Arduino y Raspberry Pi, aunque al compararlos uno a uno en cuanto a sus características y capacidades técnicas se encuentra más relacionado a las plataformas de Ardui-

no que a las de Raspberry. También, otro aspecto característico de este tipo de plataformas es que cuenta con su propio *software* y herramientas para desarrollo, aunque en el caso de las plataformas de Espressif Systems son compatibles o pueden utilizar otros entornos de desarrollo como, por ejemplo, Arduino, lo que ha ayudado a su rápida aceptación por el mercado. Tienen como base dos módulos independientes:

- *ESP8266.* Con un procesador de 32-bits RISC a 80MHz, 32 Kb de memoria RAM, conectividad Wi-Fi con varios protocolos de red y autenticación, 16 puertos de propósito general y convertidores analógico a digital de 10 bits.
- *ESP32.* Con un procesador de doble núcleo de 32 bits de hasta 240Mhz, 520 Kb de SRAM, con conectividad Wi-Fi y Bluetooth, y una muy amplia disponibilidad de puertos de entrada y salida incluyendo convertidores DAC (digital a analógico) de 8 y 12 bits, sensores táctiles, interfaces SPI, I2S, I2C y UART, entre otros, esquemas de seguridad y administración de energía.

De estos dos módulos destaca el ESP32, debido a que en su diseño ya considera el soporte para el desarrollo de aplicaciones de Inteligencia artificial de las cosas (AI-IoT), un concepto relativamente nuevo que refiere a la integración de inteligencia artificial en aplicaciones de Internet de las cosas. Aunque el uso de estas plataformas no está tan documentado como lo es en los casos de Arduino y Raspberry, sí lo están utilizando en proyectos de investigación y en aplicaciones de AI-IoT, como lo reportado por Debauch et al. (2020), Abdelouhahid et al. (2020) y Rai et al. (2019), por citar algunos.

Sensores y transductores

Debido a la revolución de la industria conectada o llamada industria 4.0, hoy contamos con una gran variedad de sensores inteligentes (*smart sensor) (*Meijer, 2008) que no solamente se dedican a leer las magnitudes de las variables en cuestión, pues también se desarrollan en concordancia con los estándares más utilizados en el contexto del IoT (Ha et al., 2020).

Los transductores son aquellos dispositivos electrónicos capaces de transformar una variable física en otra variable que es de naturaleza más sencilla de medir y manejable por los dispositivos electrónicos. Así, se les llama sensores a aquellos transductores que transforman variables físicas en variables eléctricas que son más fáciles de manipular mediante nuestros dispositivos electrónicos IoT actuales y, por ello, es el término más reconocido en los ecosistemas de IoT (Ramírez et al., 2014).

En el caso de proyectos de IoT existe una gran diversidad de sensores que, en función de las variables a medir, tienen como sus principales objetivos que estos sistemas capten la información de manera sencilla, que sean fiables y seguros, que nos ayuden en los procesos de automatización, así como que se tenga la mínima interacción con personal técnico para mantenerlos en operación, que haya una gran cantidad de información para los usuarios en cualquier momento o lugar que permita su rápida implementación y mantenimiento, que cumplan con los estándares de seguridad que imponen las áreas operacionales de TI, y que se puedan adquirir de manera rápida en el mercado en el mejor tiempo y a bajo costo adecuados a los servicios que estos prestan (Salvador, 2020).

Los sensores pueden clasificarse de varias maneras, pero la más común es por la métrica que leen, llamada magnitud. A continuación, se listan algunos ejemplos de sensores:

- Propósito general: temperatura, humedad, presión, acumuladores o de niveles de depósitos, de servicios o caudales de línea, de presencia para contar piezas, paquetes o bolsas, de tensión relacionados a los temas de consumo y potencia, de proximidad, luminosidad y ruido.
- Administración y tratamiento de aguas: PH, redox, conductividad, turbidez, oxígeno disuelto, servicios de caudales de línea, Potencial Oxidación Radiación (ORP), niveles de depósitos en estanques, manantiales y depuradoras.
- Específicos: Acelerómetro, giroscopio, para medir distancias entre elementos de una máquina o proceso.

Es importante, al utilizar sensores, conocer el tipo de señales con las cuales se va a trabajar puesto que estas pueden ser analógicas o digitales. Las analógicas toman valores entre un rango mínimo y máximo establecido como en el caso de la temperatura, que puede ser de -30 a 100 grados centígrados, por ejemplo. En el caso de las señales digitales los valores pueden estar o no presentes, es decir, activa o no activa; en la industria son muy comunes y entre ellos podemos encontrar los de presostato, termostato, de presencia, inductivos y capacitivos, switches de flujo entre otros (Thomazini y de Albuquerque, 2020).

Actuadores

Como el nombre lo indica, los actuadores son transductores que en el contexto del IoT realizan funciones de interacción desde el mundo virtual al físico, o con otros sistemas virtuales. Se refiere a dispositivos mecánicos que interactúan con ellos mismos u otros dispositivos, o también se refiere a acciones específicas que un objeto debe realizar. Entonces, los podemos dividir en dos grandes grupos:

- *Actuadores.* Son dispositivos mecánicos que convierten energía en movimiento como motores y servomotores, entre otros. Madakam et al. (2015) los clasifica en tres tipos: 1) eléctricos, como motores de corriente alterna y directa, solenoides y motores paso a paso, 2) hidráulicos, utilizan el flujo hidráulico para realizar el movimiento y 3) neumáticos, utilizan aire comprimido para realizar el movimiento.
- *Accionadores.* Son acciones que realizan dispositivos como el envío de un mensaje, el control de leds, encendido de luces o el control del movimiento de un robot (González et al., 2017).

Los actuadores son, en cierta forma, lo opuesto a los sensores. Mientras que los sensores detectan eventos o cambios en el ambiente o entorno físico, los actuadores introducen cambios en el entorno físico mediante acciones como, por ejemplo, abriendo o cerrando válvulas, cambiando la posición o ángulo de otros dispositivos, emitiendo sonidos o activando luces (Avsystem, 2020).

Conclusiones

La tecnología continúa en constante evolución, por ello la oferta de plataformas y componentes electrónicos cada vez es más diversa, posibilitando que podamos tener una amplia gama de opciones cuando se vayan a desarrollar proyectos y aplicaciones para IoT. Cada vez se cuenta con un mayor número de fabricantes que ofrecen nuevos modelos con *chips* y placas de desarrollo más potentes e incluso a precios más accesibles, con *frameworks* que integran también las herramientas para programarlos y desarrollar el *software* que nos permita controlar estos dispositivos. Esto resulta, sin duda, en una gran ventaja, pero también en un interesante reto que consiste en seleccionar la infraestructura de hardware más idónea para llevar a cabo aplicaciones de IoT.

Referencias

Abdelouhahid, R. A., Debauche, O., Mahmoudi, S., Marzak, A., Manneback, P. y Lebeau, F. (2020). Open Phytotron: a new IoT device for home gardening. En *2020 5th International Conference on Cloud Computing and Artificial Intelligence: technologies and applications (CloudTech)*, pp. 1-8. https://doi.org/10.1109/CloudTech49835.2020.9365892.

Ábrego Preza, E. S., Araujo Ayala, O. D., Góchez Zelaya, E. A. y Navarro Masferrer, C. E. (2018). *Fabricación de PLC utilizando microcontroladores para uso didáctico con características industriales*. http://redicces.org.sv/jspui/handle/10972/3392.

Avsystem. (2020). *Top sensor types used in IoT.* https://www.avsystem.com/blog/iot-sensors-iot-actuators/

Bengochea-Guevara, J. M., Espinosa-Bustillo, M. Á., Romano, G., Martín-Martínez, F. M., Muñoz-Frías, J. D. y Rodríguez Pecharromán, R. (2020). *Microprocesadores*. http://hdl.handle.net/11531/70549

Debauche, O., Abdelouahid, R.A., Mahmoudi, S., Moussaoui, Y., Marzak, A. y Manneback, P. (2020). RevoCampus: a distributed open source and low-cost smart campus. En *2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet)*, pp. 1-10. https://doi.org/10.1109/CommNet49926.2020.9199640.

El-Abd, M. A (2017). Review of embedded systems education in the Arduino age: lessons learned and future directions. *International Journal of Engineering Pedagogy, 7*(2), pp. 79-93. https://www.learntechlib.org/p/207404/.

Gómez-Guzmán, E., Mejía-Gallardo, JE., Ramírez-Guerra, LD., Cruz-Yépez, R. e Ibarra-Manzano, OG. (2021). Desarrollo de aplicaciones basadas en microcontroladores orientadas al IoT. *Revista digital Jóvenes en la Ciencia*, 10. https://www.jovenesenlaciencia.ugto.mx/index.php/jovenesenlaciencia/article/view/3349

González García, C., Meana-Llorián, D., Pelayo García-Bustelo, B. y Cueva Lovelle, J. (2017). A review about Smart Objects, Sensors, and Actuators. *International Journal of Interactive Multimedia and Artificial Intelligence*, 4, pp 7-10. https://doi.org/10.9781/ijimai.2017.431.

Ha, N., Xu, K., Ren, G., Mitchell, A. y Ou, J. Z. (2020). Machine learning-enabled smart sensor systems. *Advanced Intelligent Systems*, *2*(9), 2000063.

Hunkeler, U., Truong, H. L. y Stanford-Clark, A. (2008). MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks. *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*, pp. 791-798.

Hurtuk, J., Chovanec, M. y Adam, N. (2017) The Arduino platform connected to education process. En *2017 IEEE 21st International Conference on Intelligent Engineering Systems (INES)*, pp. 000071-000076. https://doi.org/10.1109/INES.2017.8118531.

IoTConsulting.tech. (2019, 6 de diciembre). Cómo elegir un microcontrolador para IoT. https://iotconsulting.tech/como-elegir-un-microcontrolador-para-iot/

Lee, E. (2020). A meta-analysis of the effects of Arduino-based education in korean primary and secondary schools in engineering education. *European Journal of Educational Research, 9*(4), pp. 1503-1512. https://https://doi.org/10.12973/eu-jer.9.4.1503

María, F. y Daniel, J. (2018). *Microprocesadores*.

Madakam, S., Ramaswamy, R. y Tripathi, S. (2015) Internet of Things (IoT): a literature review. *Journal of Computer and Communications*, *3*(3), pp. 164-173. https://doi.org/10.4236/jcc.2015.35021

Mahmood, S., Palaniappan, S., Hasan, R., Sarker, K. U., Abass, A. y Rajegowda, P. M. (2019). Raspberry PI and role of IoT in Education. En *4th MEC International Conference on Big Data and Smart City (ICBDSC)*, pp. 1-6. https://doi.org/10.1109/ICBDSC.2019.8645598.

Meijer, G. (2008). *Smart sensor systems*. John Wiley *&* Sons.

Ramírez, L. G. C., Jiménez, G. S. A. y Carreño, J. M. (2014). *Sensores y actuadores*. Grupo Editorial Patria.

Salvador, D. *(2020). Sensores IoT: características y aplicaciones. Nespra Smart Devices*. https://www.nespra.net/blog/sensores-iot-caracteristicas-y-aplicaciones/

Thomazini, D. y de Albuquerque, P. U. B. (2020). *Sensores industriais: fundamentos e aplicações*. Saraiva Educação SA.

Valdés, F. y Areny, R. P. (2007). *Microcontroladores fundamentos y aplicaciones con PIC Vol. 1149*. Marcombo.

Yamanoor, N. S. y Yamanoor, S. (2017). High quality, low cost education with the Raspberry Pi. En *2017 IEEE Global Humanitarian Technology Conference (GHTC)*, 2017, pp. 1-5. https://doi.org/10.1109/GHTC.2017.8239274.

Capítulo 4

Conectividad IoT

Juan Manuel Ramírez Alcaraz
Ma. Guadalupe Álvarez Negrete
Arturo Cano Rueda
Carlos Alberto Flores Cortés

Resumen

El propósito final del uso de los dispositivos de Internet de las cosas (IoT) es el de proporcionar un entorno de interactividad automatizada entre las personas, las cosas y en general el medio ambiente en el que se desarrollan. Para lograr esto, es necesario que tales dispositivos sean accesibles y manipulables desde una computadora o un dispositivo móvil, lo cual implica necesariamente que el dispositivo IoT debe contar con alguna tecnología de comunicación que permita conectarlo a una red de datos.

En este capítulo se abordan los conceptos y tecnologías comunes con que se establece comunicación con los dispositivos IoT.

Conexión de dispositivos IoT

En general, un escenario de IoT involucra sensores, actuadores, controladores y una capa de *software* que establece cómo será la interacción entre ellos. Este escenario, a su vez, tendrá invariablemente una conexión a una red que permita monitorizar e interactuar con las variables físicas involucradas desde un dispositivo de cómputo y almacenar datos para su posterior procesamiento.

Aunque es posible encontrar implementaciones de IoT que utilicen medios cableados para conectarse en red (como los basados en Raspberry Pi, que incluyen una interfaz Ethernet), el medio utilizado generalmente es el inalámbrico. Existen diversas tecnologías inalámbricas que se utilizan para implementar escenarios de IoT, entre las cuales podemos mencionar a Bluetooth, NFC, Z-Wave, Zigbee, Wi-Fi y celulares.

Podemos distinguir tres tipos de conexión general dependiendo del nivel de integración del dispositivo IoT:

1. Conexión con dispositivos integrados y app de administración.
2. Conexión con dispositivos integrados y registro en un servidor.
3. Conexión con elementos integrables.

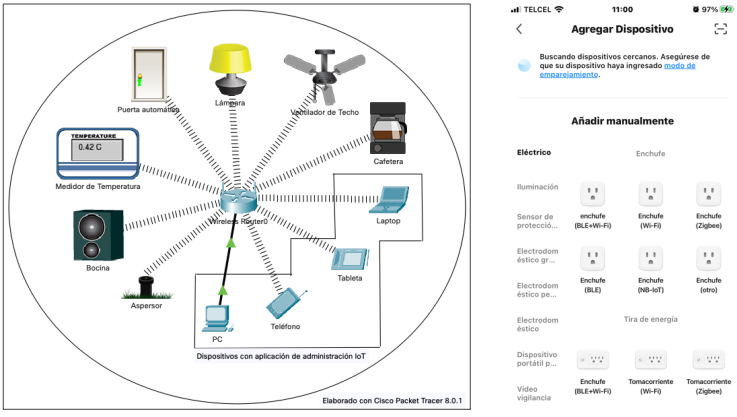
En el primer tipo de conexión, los dispositivos cuentan con un sistema integrado que puede incluir sensores, actuadores, microcontrolador, módulo de conexión inalámbrica y *software*, por ejemplo, una lámpara, una cafetera, aspersor, cámara, etc. Estos tipos de dispositivos IoT generalmente se conectan de manera directa con un punto de acceso inalámbrico, funcionando como un nodo más dentro de la red. El *software* integrado en el dispositivo incluye generalmente un servidor web que está a la escucha permanente de solicitudes de conexión en un puerto específico.

La interacción con estos dispositivos se lleva a cabo mediante una aplicación desde un equipo de cómputo, comúnmente un dispositivo móvil conectado a la red. Por medio de esta aplicación se podrán descubrir los dispositivos en su entorno y registrarlos para su manipulación o registrar usuarios que previamente hayan bajado también la aplicación y creado una cuenta en dicho sistema. El *software* usualmente es proporcionado por el mismo fabricante de los dispositivos IoT, pero existen también aplicaciones genéricas, como Smart Life proporcionada por Apple, que pueden reconocer dispositivos de distintas tecnologías.

Una vez que se han registrado los dispositivos o usuarios en la aplicación, la interacción con el dispositivo IoT se puede realizar desde cualquier lugar donde haya conexión a Internet. En la figura

1 se muestra un diagrama de una red con conexión de dispositivos integrados.

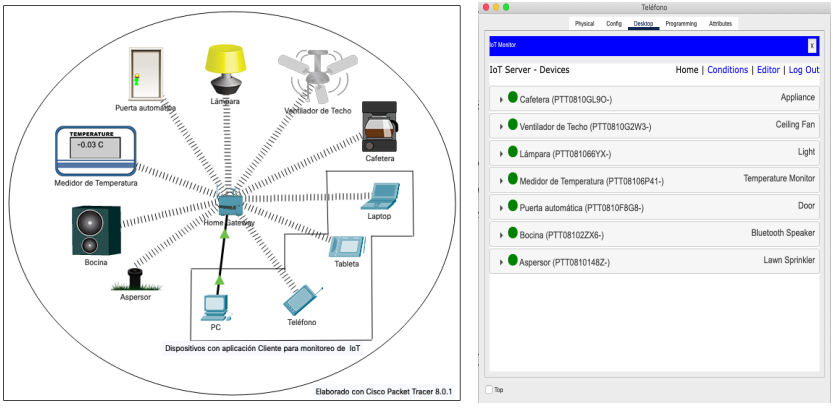
Figura 1. Escenario IoT con dispositivos integrados y aplicación de administración



Fuente: Elaboración propia.

Otra opción de interacción es mediante la implementación de un servidor en el cual se registran los dispositivos IoT y los clientes se conectan al servidor para monitorizarlos mediante una aplicación. El elemento central en esta topología recibe el nombre *Home Gateway* y es quien se encargará de intermediar entre los dispositivos de usuario con los dispositivos de IoT, típicamente este dispositivo cuenta con conexión a Internet para permitir la manipulación remota de los dispositivos IoT locales. La figura 2 muestra un diagrama general de este tipo de escenario.

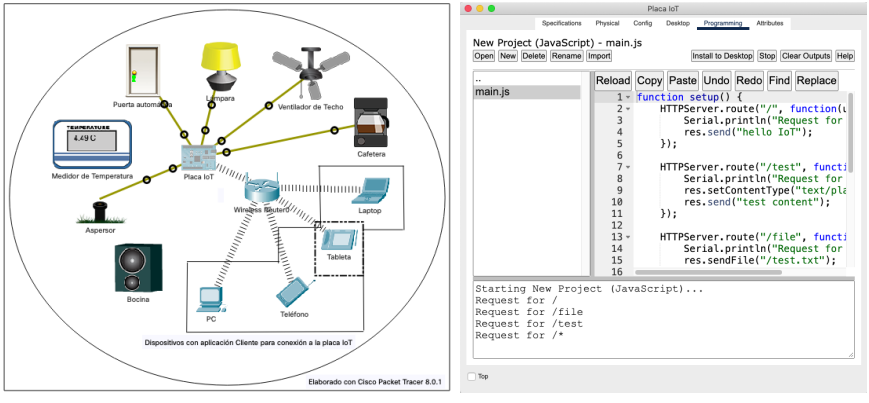
Figura 2. Escenario IoT con dispositivos integrados y registro en un servidor



Fuente: Elaboración propia.

Una tercera opción de conexión de dispositivos IoT es la de utilizar elementos integrables (ver figura 3), conectados a una placa de desarrollo. Esta opción es más recomendable para desarrolladores de apps y de escenarios de IoT, puesto que permite un control más flexible de los elementos involucrados.

Figura 3. Escenario IoT con dispositivos integrables



Fuente: Elaboración propia.

Existe una gran variedad de placas con diferentes características, entre ellas se pueden mencionar: Arduino, NodeMCU, Blue Pill, Curiosity Nano, Argon Particle basadas en microcontrolador, y Raspberry Pi, BeagleBone basadas en microprocesador, lo que las convierte en una computadora completa. Estas placas pueden conectarse a la red de datos, con lo cual existe la posibilidad de interactuar con los dispositivos IoT a través de ellas, desde cualquier lugar con conexión a Internet.

Algunas placas pueden programarse para implementar un servidor web que esté escuchando peticiones de clientes HTTP, y de esta manera estar enviando información de los diferentes sensores que tenga conectados; asimismo, las personas pueden interactuar con la placa mediante su página web para indicarle acciones a realizar, por ejemplo, prender o apagar un foco.

Una vez implementado el escenario IoT, también es posible utilizar un servicio en la nube para interactuar con los dispositivos IoT. Estos servicios incluyen almacenamiento de datos, graficación, análisis, seguridad de acceso, entre otros. Ejemplos de empresas que ofrecen estos servicios son Ubidots y Particle.

Tecnologías de redes de corto alcance y bajo consumo

Para la implementación de escenarios de IoT es importante considerar la tecnología de comunicación que tienen implementada los dispositivos IoT que se utilizarán, puesto que aun cuando existen aplicaciones que pueden trabajar con dispositivos de diferentes tecnologías, se puede dar incompatibilidad entre ellos.

En esta sección se describen las principales tecnologías de comunicación que se implementan en los dispositivos IoT en un entorno local.

Bluetooth Low Energy (BLE)

Bluetooth

Bluetooth Low Energy (BLE) (Bluetooth SIG, 2021) es una versión del Bluetooth pensada para funcionar con muy baja potencia pero conservando varias características del Bluetooth clásico.

Transmite en la frecuencia de 2.4 GHz y tiene un alcance que normalmente se acota a 10 metros; sin embargo, con la combinación adecuada de parámetros, Bluetooth puede alcanzar más de 1 km dependiendo diferentes factores como: el rango de frecuencias, la Capa Física, sensitividad del receptor, la potencia de transmisión, la ganancia de la antena y la pérdida de la señal. Es la implementación de los fabricantes para los casos de uso la que ha determinado el alcance actual. La velocidad de transferencia, cuando se utiliza la capa física (PHY) LE 2M, puede llegar a alcanzar 1.4 Mbps en el nivel de aplicación.

BLE, a diferencia de la versión clásica, puede implementarse utilizando las topologías punto-a-punto, *broadcast* y, desde 2017, en Malla. Esta tecnología también permite conocer si un dispositivo se encuentra dentro de un radio determinado (proximidad) y la dirección hacia donde este se encuentra. Próximamente también será posible medir la distancia.

Su enfoque principal es en aplicaciones que no necesiten intercambiar grandes cantidades de datos, lo que, aunado a sus requerimientos de baja potencia (los dispositivos pueden estar en uso durante meses o años sin cambiar la batería) y su bajo costo, la hace adecuada para su uso en IoT. Actualmente existe una gran variedad de casos de uso de BLE para IoT, en donde se cuentan aplicaciones en el sector automovilístico, la seguridad, los sensores de proximidad, los dispositivos para el hogar y los dispositivos para el cuidado de la salud personal. Un ejemplo de ellos es un rastreador, que puede adecuarse para localizar a una persona, una mascota, las llaves en la casa, un vehículo en el estacionamiento, una maleta en el aeropuerto, etc. (Celer, 2020)"languag e":"es","title":"Un Truco para Facilitar el Reclamo de Equipaje en el Aeropuerto","URL":"https://www.celersms.com/baggage-claim-es.htm","author":[{"literal":"Victor Celer"}],"accessed":{"date-parts":[["2021",12,13]]},"issued":{"date-parts":[["2020"]]}}],"schema":"https://github.com/citation-style-language/schema/raw/master/csl-citation.json"} . Otro caso de uso, descrito en Swedberg (2021), es el de un desfibrilador automático que proporciona alertas e información para actuar rápido en caso de un paro cardiaco.

NFC

La *Near Field Communication* (NFC) (NFC Forum, 2021) es una tecnología inalámbrica pensada para funcionar en 13.56 MHz. Su velocidad de transferencia puede llegar a los 424 kbit/s. Está diseñada para operar en distancias de unos pocos centímetros, lo que la hace también segura puesto que es difícil que un atacante pueda registrar la comunicación. No necesita emparejamiento previo como Bluetooth. NFC es una tecnología de bajo consumo de energía.

Los elementos básicos de esta tecnología son los dispositivos NFC y las etiquetas NFC. Los dispositivos generan un campo RF (*Radio Frequency*) por medio del cual pueden transferir energía a una etiqueta o enviar y recibir información de ella. Las etiquetas son elementos pasivos que no necesitan baterías para funcionar, lo cual las hace ideales para su aplicación en dispositivos IoT pequeños.

Dependiendo el caso de uso, la NFC se puede utilizar en seis formas diferentes:

1. Escritura/lectura. El caso más común, en donde un dispositivo se comunica con una etiqueta.
2. *Peer to Peer* (P2P). Dos dispositivos intercambian datos, por ejemplo, para intercambiar los datos de contacto de los usuarios de los dispositivos.
3. Emulación de tarjeta. El dispositivo opera como una tarjeta capaz de comunicarse con un lector. El caso de uso típico es la emulación de tarjetas bancarias para realizar transacciones monetarias.
4. Emulación de tarjeta de host (HCE-*Host Card Emulation*). Transfiere información entre un dispositivo NFC y una aplicación en un dispositivo móvil que emula una tarjeta NFC.
5. Carga inalámbrica. Un dispositivo NFC transfiere carga a algún dispositivo IoT pequeño de hasta 1W.
6. Emulación de tarjeta basada en elemento seguro. La emulación de tarjeta se provee mediante un elemento seguro dentro del dispositivo, como puede ser un *chip* o una tarjeta SIM.

Con NFC se soluciona el problema de objetos sin alimentación que no tienen acceso a la red, ya que puede agregar inteligencia en cualquier lugar, solo insertando etiquetas NFC a objetos desconectados y sin fuentes de energía.

Usos comunes de la tecnología NFC en IoT:

- Identificación.
- Recogida/intercambio de datos.
- Sincronización instantánea de dispositivos.
- Automatización de acciones.
- Pago con el teléfono móvil.

Zigbee

Zigbee (Connectivity Standards Alliance, 2021) es una de las mejores soluciones inalámbricas de bajo consumo de energía para escenarios IoT locales. Es gestionado por la Connectivity Standards Alliance y es un estándar abierto, por lo que tiene una gran expansión y flexibilidad.

Utiliza una topología de malla, por lo que requiere de un Hub o controlador, que es quien finalmente provee la interacción con Internet. La comunicación entre los nodos puede dar tantos saltos como sea necesario hasta llegar al Hub.

Su rango de alcance promedio es de hasta 300 metros en línea de vista, y entre 75 y 100 metros en interiores. Soporta teóricamente hasta 65,000 nodos, con una velocidad de transferencia que va de 40 a 250 kbps en la banda de 2.4 GHz. La versión 3.0 garantiza la interoperabilidad de productos de distintos fabricantes.

ZigBee gestiona la red de manera automática, pues tiene la capacidad de enrutar los dispositivos dinámicamente, cambiando por sí solo las rutas cuando un nodo nuevo entra a la red o sale de ella.

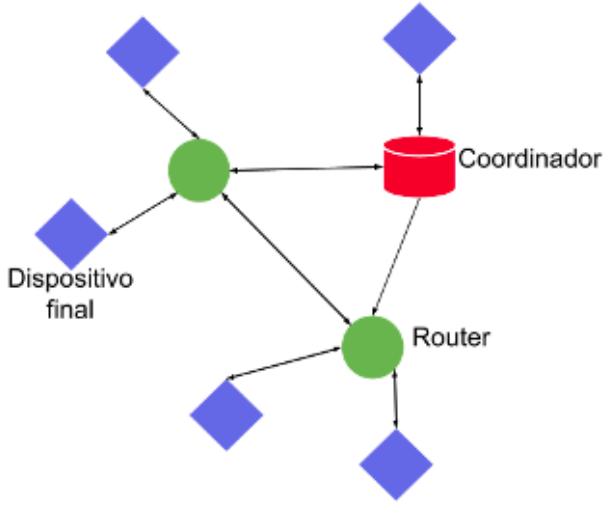
Un nodo dentro de una red ZigBee puede tener los siguientes roles (Venco Electrónica, 2020):

- Coordinador: es un nodo único y el que maneja la red, se encarga del enrutamiento y verificación de los nodos entrantes y salientes.

- *Router*: como el coordinador, pero este solamente se encarga de enrutar los dispositivos de la red.
- Dispositivo final: solamente envían o reciben paquetes de la red, pueden además activar su modo *sleep* cuando no estén activos y volverse a activar cuando se necesite, esto para reducir el gasto de energía.

En la figura 4 se muestra un diagrama básico de la interacción de estos elementos.

Figura 4. Red de malla Zigbee



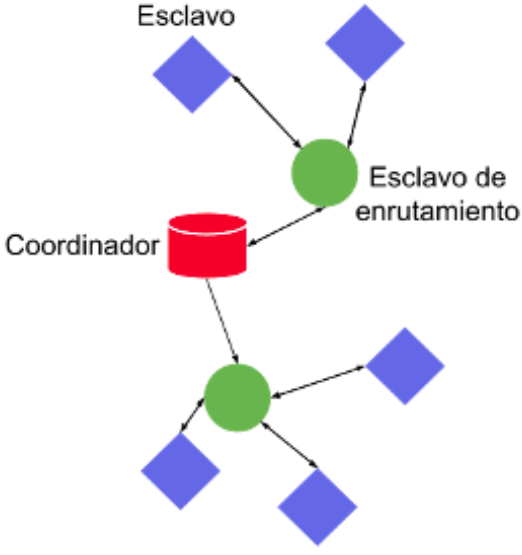
Fuente: Elaboración propia.

Actualmente es posible encontrar una gran variedad de dispositivos que utilizan Zigbee: enchufes, sensores, alarmas, *relays*, medidores de consumo de energía, entre otros (Develco Products, 2021).

Z-Wave

Z-Wave (Z.Wave Alliance, 2021) es un estándar propietario cuyo dueño es Silicon Labs. Es una de las tecnologías más utilizadas en productos para casas inteligentes. Al igual que ZigBee, se basa en una topología de red en malla (ver figura 5).

Figura 5. Red de malla Z-Wave



Fuente: Elaboración propia.

Cada dispositivo instalado en la red se convierte en un repetidor de señal, por lo que, cuantos más dispositivos se tenga, más fuerte se volverá la red. Incluso varias redes Z-Wave se pueden conectar entre sí para implementaciones aún más grandes. La señal tiene un alcance de 100 metros en ambientes abiertos, pero se recomienda tener un dispositivo cada 9 m. Cada red puede admitir hasta 232 dispositivos, con una velocidad de 40 a 100 kbps. No hay interferencia con la señal de Wi-Fi ya que opera en frecuencias por debajo de 1 GHz.

Z-Wave solo permite dar cuatro "saltos" de dispositivo como máximo para llegar al coordinador también conocido como Smart Home Hub (*Smart Hubs*, 2021). Otro inconveniente es su precio, Z-Wave es una tecnología de gama alta, con un precio elevado.

Wi-Fi/802.11

La tecnología Wi-Fi se basa en el estándar de comunicación inalámbrica 802.11 desarrollado por el grupo de trabajo IEEE 802.11 WLAN (IEEE 802.11 WLAN Working Group, 2021), que ha ido mejorando constantemente aumentando la velocidad y reduciendo la latencia. Existe específicamente una Wi-Fi Alliance (Wi-Fi Alliance, 2021a) integrada por compañías de todo el mundo que colaboran para proveer un uso interoperable, seguro y confiable de Wi-Fi. Esta organización ha establecido nombres generacionales para hacer más fácil comprender las diferentes versiones de Wi-Fi. La tabla 1 muestra la relación de estos nombres.

Tabla 1. Relación de nombres de las versiones Wi-Fi

Wi-Fi 7	802.11be (en desarrollo)
Wi-Fi 6	802.11ax
Wi-Fi 5	802.11ac
Wi-Fi 4	802.11n
Wi-Fi 3	802.11g
Wi-Fi 2	802.11b
Wi-Fi 1	802.11a

Fuente: Elaboración propia.

Cabe señalar que las versiones 1, 2 y 3 en realidad no se consideran oficiales por ser relativamente antiguas. Wi-Fi 6 es retrocompatible, por lo tanto, un dispositivo Wi-Fi 6 puede conectarse a redes Wi-Fi 4 o 5.

Wi-Fi 6 puede operar tanto en la frecuencia de los 2,4 GHz como en la de los 5 GHz, pero existe una nueva extensión de Wi-Fi 6 llamada Wi-Fi 6E con la cual se agrega la banda de los 6 GHz (Wi-Fi Alliance, 2021b). Las principales diferencias entre 2.4GHz y 5GHz son la velocidad de transferencia y el alcance, la primera tiene mayor alcance, pero menos velocidad que la segunda. Por otro lado, 2.4GHz lleva más tiempo usándose, por lo tanto, es compatible con casi cualquier dispositivo.

2.4GHz
Alcance: 45 metros en interiores y 90 metros en exteriores.
Velocidad: alrededor de 60 Mbps.

5GHz
Alcance: 15 metros en interiores y 30 metros en exteriores.
Velocidad: alrededor de 867 Mbps.

Wi-Fi 6 promete una mejor eficiencia energética al incluir la tecnología *Target Wake Time*, o TWT. De este modo, un dispositivo puede ponerse en modo de ahorro energético hasta que sea el momento en el que debe conectarse para recibir nuevos datos (Ramírez, 2021).

Wi-Fi Direct y Wi-Fi Aware

Estas son dos tecnologías basadas en el estándar Wi-Fi que permiten la conexión directa de dispositivos sin necesidad de tener una red de infraestructura, conexión a Internet o señal de GPS, lo que abre una amplia gama de aplicaciones en diversos ámbitos incluido IoT. Algunos ejemplos de aplicación son la conexión entre gente cercana para jugar, escuchar canciones o compartir fotos, enviar documentos directamente a la impresora, ver el menú de un restaurante, entre muchos otros (Wi-Fi Alliance, 2021c).

Wi-Fi también puede implementarse en placas de desarrollo para IoT, lo que permite que la placa tenga la posibilidad de conectarse en red con un punto de acceso y, si el módulo Wi-Fi tiene la tecnología Aware o Direct, será posible enlazarse directamente con otros dispositivos compatibles.

Redes de área extensa de bajo consumo (LPWAN)

Las redes de área amplia de baja potencia (redes LPWAN - LPWA: *Low Power Wide Area Network*) son un conjunto de tecnologías de comunicación inalámbrica diseñadas para comunicaciones de bajo costo, de largo alcance y de bajo consumo de energía desde dispositivos IoT simples.

Gracias al creciente apoyo de políticas y plataformas, las tecnologías de Internet de las cosas se están desarrollando rápidamente con diversos tipos de tecnología en crecimiento. Sin embargo, con la evolución y las necesidades de comunicación individuales han aumentado algunos tipos en particular (Hossain y Markendahl, 2021), tal es el caso de las conexiones de comunicación IoT de bajo consumo, de área amplia y de largo alcance, llegando a convertirse en la actualidad en una tendencia demandante.

Algunas de las tecnologías que usualmente utilizamos en la vida diaria llegan a tener algunas desventajas, como el alto consumo de energía y, sobre todo, el costo. Por ello, para satisfacer la necesidad de conexiones de comunicación de IoT de largo alcance, han surgido las LPWAN (*Low Power Wide Area Networks*, Redes de área extensa de bajo consumo).

La mayoría de las LPWAN logran un extenso alcance de comunicación y forman una topología en estrella en la que los dispositivos tienen comunicación directamente con la estación base. Algunas LPWAN no celulares operan en bajas frecuencias (banda sub-GHz[1]) las cuales brindan un amplio rango de comunicación, que van desde cientos de metros en interiores y varios kilómetros en exteriores.

Las frecuencias más bajas tienen una mejor característica de propagación a través de obstáculos, por lo que estas cualidades hicieron todavía más atractiva a la banda sub-GHz implementada en las tecnologías LPWAN. En contraste con las redes inalámbricas de sensores de corto alcance (WSN), el objetivo de diseño de las LPWAN es ofrecer una amplia cobertura de baja potencia y, por lo tanto, acceder a ellas a través de un costo más bajo. Las LPWAN

[1] Frecuencias de banda menores a 1 GHz que difieren dependiendo en el país o región en que se encuentren.

logran un funcionamiento de baja potencia utilizando varios enfoques, los cuales se describen a continuación:

- Forman una topología en estrella, la cual elimina la energía consumida a través del enrutamiento de paquetes en red multisalto.
- Mantienen un diseño simple del nodo al descargar las complejidades a la estación base o puerta de enlace.
- Utilizan canales de banda estrecha, lo que reduce el nivel de ruido y, en consecuencia, se extiende el rango de transmisión.

De acuerdo con la banda de frecuencia operativa, las tecnologías de comunicación aplicadas se pueden dividir en dos tipos: las que operan en un espectro sin licencia y las que operan en un espectro autorizado. En el primer tipo podemos mencionar la tecnología Lora (explicada más adelante) y Sigfox (en párrafos siguientes). En el segundo se clasifican todas las tecnologías compatibles con el protocolo 3GPP como, por ejemplo, NB-IoT que se construyen con base en redes móviles licenciadas.

5G para IoT

El desarrollo de la tecnología de comunicación 5G promete satisfacer las necesidades de arquitecturas complejas de IoT. La expectativa de la nueva generación de comunicación 5G es proporcionar una velocidad entre 10-800 Gbps, comparando este número con la tecnología actual 4G con una velocidad de 2-1,000 Mbps, 5G debe poder manejar el tráfico producido por los dispositivos IoT.

También se espera que la tecnología 5G se adapte tanto a IPv4 como a IPv6, por tener traducción de *framework* para ambas. La implementación de 5G se define mediante muchas tecnologías actuales y en desarrollo, tales como: redes heterogéneas (HetNets), redes definidas por *software* (SDN), MIMO masivo y acceso múltiple por radio, etc. Sin embargo, todas estas tecnologías vienen con sus propios desafíos de seguridad. Por ejemplo, HetNets tendrá un traspaso frecuente que afecta directamente el proceso de autenticación en la red, especialmente con el requisito de latencia de

5G. Además, la computación en la nube y las SDN aumentarán la cantidad de ataques DDoS debido a la característica de autoservicio bajo demanda de la computación en la nube. La seguridad de 5G y todas las tecnologías emergentes involucradas en ella deben abordarse de manera extensa, a fin de garantizar la seguridad de IoT (Baraza, 2019)"author":[{"family":"Baraza","given":"Alexander de Jesús Celín"}],"issued":{"date-parts":[["2019"]]}}],"schema":"https://github.com/citation-style-language/schema/raw/master/csl-citation.json"} .

El inconveniente de la arquitectura 4G es que puede admitir un ancho de banda máximo de 1 Gigabit y, a medida que aumenta el ancho de banda requerido por los dispositivos IoT, 4G puede convertirse en un cuello de botella. Por otro lado, 4G puede ser vulnerable a los piratas informáticos y los virus. Como la seguridad de los datos y el ancho de banda son más importantes para los dispositivos IoT, 4G pronto no será una tecnología adecuada para su uso.

En comparación con los sistemas 4G, los sistemas celulares 5G trabajan en frecuencias más altas donde es más fácil obtener anchos de banda mucho más amplios.

La tecnología 5G se basa en:

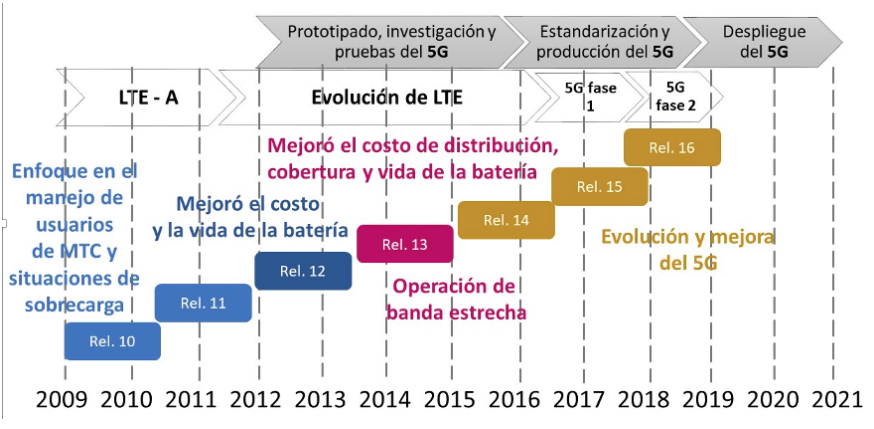
- Acceso de radio económico que alcanza velocidades de datos superiores a 10 Gb/s, a través de la comunicación de bandas de frecuencia más altas por encima de 6 GHz y tecnologías relacionadas.
- Virtualización de Funciones de Red (NFV), la cual permitirá implementar funciones de red específicas en *software* que se ejecuta en *hardware* genérico sin la necesidad de costosas máquinas específicas de *hardware*. Con esto, se tiene la reducción de costos de implementación, administración y operación al permitir reutilizar y compartir la misma funcionalidad entre clientes, un ejemplo es la plataforma Docker[2].

[2] Plataforma que despliega aplicaciones dentro de contenedores de *software*, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos, en especial para IoT.

Las redes definidas por *software* (SDN) permitirán que el control de los recursos de la red se abra a terceros, teniendo la flexibilidad para acomodar aplicaciones exigentes a nivel profesional.

En la figura 6 se muestra una línea de tiempo con la evolución de las tecnologías LTE y 5G.

Figura 6. Línea del tiempo de tecnologías 3GPP y sus lanzamientos



Fuente: Elaboración propia, basada en Palatella et al. (2016).

LoRaWAN

LoRaWAN utiliza técnicas de modulación lineal de espectro ensanchado, que no solo mantiene la característica de bajo consumo de energía, sino que también aumenta significativamente la distancia de comunicación. Debido a ello, los dispositivos finales con diferentes secuencias de espectro ensanchado no interferirán entre sí, incluso sin transmitir usando la misma frecuencia al mismo tiempo. Por lo tanto, los dispositivos desarrollados bajo este esquema pueden recibir y procesar datos de varios nodos en paralelo, lo que puede ampliar enormemente la capacidad del sistema.

LoRa usa bandas ISM (Industrial, Scientific and Medical) sin licencia, pero se usan diferentes frecuencias de este tipo en algunos países y regiones. En el mercado chino, China LoRa Applica-

tion Alliance (CLAA), liderada por ZTE, generalmente recomienda el uso de la banda 470-518 MHz, pero el uso de la banda 470-510 MHz para radiómetros. Dado que LoRa opera en bandas sin licencia, no requiere ninguna aplicación para la construcción de la red y su arquitectura es simple, además del costo que es más económico (Zhiwei y Jie, 2021).

LoRaWAN está adaptado y modula en la banda de SubGHz sin licencia. Para modulación, utiliza las técnicas patentadas de *chirp*[3] de espectro ensanchado (CSS). LoRaWAN comparte la banda ISM 868 MHz en Europa y 915 MHz en América del Norte junto con Sigfox. Lora, también admite la transmisión bidireccional limitada de una señal de banda estrecha sobre un canal de banda ancha. Utiliza factores de dispersión (SF) que dan un intercambio entre área de cobertura extendida y ancho de banda; asimismo, dependiendo de diferentes SF, la velocidad de datos puede variar entre 300 bps y 50 kbps. Esto significa que el usuario del borde de la celda puede transmitir a 300 bps y, cuanto más cerca esté un dispositivo del punto de acceso, mayor rendimiento puede lograr.

Arquitectura de red LoRaWAN

LoRaWAN es un conjunto de protocolos de comunicación y arquitectura de sistema diseñados con base en comunicaciones de largo alcance LoRaRed, la cual puede proporcionar una conexión de red regional, nacional o global para varios dispositivos inalámbricos alimentados por batería.

Debido a su alta escalabilidad y compatibilidad para el desarrollo de aplicaciones integradas, se pueden lograr conexiones casi perfectas sin una configuración compleja. Según las clases de protocolos, LoRaWAN es capa MAC y LoRa es la capa física. Debido a esto, LoRaWAN originalmente se llamaba LoRaMAC. Este protocolo se ha optimizado para sensores de bajo consumo de energía y alimentados por batería. Cubriendo diferentes niveles de nodos de dispositivos finales, el equilibrio entre la latencia de la red y la duración de la batería ha sido optimizado. Para ello se ha construido un protocolo

[3] Un *chirp* consiste en un especial fenómeno oscilante caracterizado por rápidas variaciones de frecuencia, presente en una señal.

básico en LoRaWAN, en el que los fabricantes globales en el campo de IoT pueden desarrollar, producir y fabricar productos basados en él, siendo más rentable en comparación con otras redes.

Los elementos básicos que componen una red LoRaWAN son: nodos o dispositivos finales, *gateways* o puertas de enlace, servidores de red y servidores de aplicación. La red LoRa está organizada en una topología de estrella especial, llamada estrella de estrellas, en la cual las puertas de enlace retransmiten mensajes entre los nodos finales y un servidor de red. Los nodos generalmente están conectados a sensores o a actuadores, recopilan datos y los transmiten a un *gateway* a través del protocolo LoRaMAC.

El gateway LoRa puede proporcionar más de 20,000 conexiones de dispositivos finales. La distancia de cobertura de una sola puerta de enlace suele ser de 3 a 5 km, lo que puede superar la de las redes celulares tradicionales en entornos urbanos algo complejos. En áreas abiertas, su distancia de cobertura puede ser incluso de hasta 15 km.

El *gateway* no procesa datos, solo empaqueta y encapsula los datos recibidos de los nodos finales, después los transmite únicamente al servidor. La comunicación del dispositivo final al *gateway* se conoce como *enlace ascendente* y la comunicación del *gateway* al dispositivo final se conoce como *enlace descendente*.

El servidor de red recibe los datos de las puertas de enlace y realiza un procesamiento sobre ellos. Enrutamiento, verificación de errores en trama, dar respuesta a requerimientos de nodos finales, son algunas de las actividades que realiza el servidor de red. El servidor de aplicación procesa la información útil de los mensajes enviados por los nodos finales y, generalmente, provee esta información a los usuarios finales.

La transmisión del protocolo LoRaWAN tiene tres modos, que son: Clase A, Clase B y Clase C. Estas clases ofrecen una compensación entre comunicación de enlace descendente, latencia y eficiencia energética (vida útil de la batería).

- Clase A: son comunicaciones bidireccionales donde cada transmisión de enlace ascendente es seguida de dos ranuras de recepción de enlace descendente cortas dependiendo de la necesidad de aplicación. El dispositi-

vo final programa aleatoriamente las ranuras del enlace descendente basado en un protocolo similar a ALOHA[4]. El dispositivo final solo puede recibir después de enviar, esto es, no hay restricción en el enlace ascendente, y los datos del enlace descendente solo pueden ser recibidos por el dispositivo final después de que se envía el paquete de enlace ascendente, esto hace que se logre un menor consumo de energía.

- Clase B: el dispositivo final negocia con el servidor sobre el momento en que la ventana de recepción es abierta y luego recibe a la hora acordada. Se pueden recibir varios paquetes a la vez. Segundo menor consumo de energía.
- Clase C: el dispositivo final abre la ventana de recepción en todo momento, excepto cuando envía. El cual consume más energía, pero tiene un retardo de comunicación más bajo (poder más alto consumo), es por eso que, en esta clase, la ventana de recepción está continuamente abierta, a menos que el dispositivo final esté transmitiendo.

Para asegurar la transmisión de paquetes confiable, así como optimizar el rendimiento de la red y la capacidad del nodo de los dispositivos finales, el servidor LoRaWAN implementa el algoritmo *Adaptive Data Rate* (ADR) para controlar la transmisión de datos índice. Actualmente, muchas redes implementadas adoptan una arquitectura de malla, en la que cada nodo de dispositivos finales puede reenviar información desde otros nodos, actuando como una función de enrutamiento. Al hacerlo, aumentan la distancia y el alcance de la comunicación en la red.

Por otro lado, la arquitectura malla aumenta la complejidad del sistema y reduce la capacidad de la red. Además, para recibir y procesar estos mensajes, la batería tiene un alto consumo de energía, lo que acortará su vida útil, mientras que de estos mensajes, solo una pequeña parte está relacionada con el nodo, lo que signi-

[4] Protocolo en el cual cada vez que una estación tiene datos para enviarlos, los transmite sin esperar, mientras que en ALOHA un usuario espera hasta la próxima vez que coincida su marco de datos.

fica que la mayor parte del tiempo y el poder del nodo se desperdicia procesando mensajes irrelevantes. En LoRaWAN, se adopta una estructura en estrella de largo alcance, que no solo permite una conexión igualmente amplia, sino que también extiende el tiempo de uso de la batería, ahorrando así costos.

Lora utiliza la técnica de modulación FHSS (*Frecuency Hopping Spread Spectrum*-Espectro Extendido por Salto de Frecuencia), lo cual permite disminuir la interferencia ocasionada por las señales inalámbricas que pueden existir en el entorno.

Narrow Band -IoT (NB-IoT)

La NB-IoT, también conocida como LTE CAT-M2, opera en Banda LTE. Con NB-IoT no se requieren puertas de enlace, por lo que los datos del sensor se transfieren directamente al servidor principal. CAT-M2 tiene menor costo y consumo de energía que CAT-M1. Debido a estas características, ha ganado popularidad para proyectos de ciudades inteligentes. La potencia máxima transmitida es de 20 dBm por encima de 1.4 MHz de ancho de banda. La complejidad se reduce en un 75% en comparación con el CAT-M1. Las velocidades de datos son de alrededor de 200 kbps con una latencia de 1.6 a 10 segundos.

NB-IoT tiene una sólida capacidad de cobertura en interiores, que es aproximadamente 20 dB más alta que LTE, equivalente a 100 veces más en términos de cobertura inalámbrica. Satisface no solo la demanda de una amplia cobertura sino también la de cobertura profunda requerida para aplicaciones en fábricas, garajes subterráneos y tapas de alcantarilla, cuya distancia de transmisión suele ser superior o igual a 10 km.

NB-IoT utiliza bandas ISM con licencia y un número de banda definido por LTE. La versión 13 especifica 14 bandas para su uso. Actualmente, las principales bandas NB-IoT en todo el mundo son 868 MHz y 915 MHz. Narrowband-IoT está estandarizado en 3GPP versión 13 y puede implementarse en banda con licencia (en banda independiente) y banda sin licencia. En la banda con licencia no hay limitaciones en el ciclo de trabajo. En la banda sin licencia, el ciclo de trabajo depende de la política de regulación del espectro de la región específica.

En algunas ocasiones la puerta de enlace LoRa puede funcionar a una distancia de 100 km. Dado que NB-IoT se basa en redes celulares y ocupa solo alrededor de 180 KHz de ancho de banda, se despliega directamente en las distintas redes de comunicaciones que establecen los operadores de telecomunicaciones.

Con la misma estación base, NB-IoT puede aumentar el número de accesos entre 50 y 100 veces en comparación con las tecnologías de comunicación existentes, y un sector puede soportar 100,000 conexiones. La nueva radio NB-IoT optimiza la eficiencia y la cobertura de la batería, pudiendo soportar una vida útil prolongada de hasta aproximadamente 15 años.

NB-IoT se extenderá para incluir servicios como localización y multidifusión en su próxima versión.

Sigfox

SigFox es una tecnología LPWAN basada en la técnica de modulación de banda ultra estrecha (UNB por sus siglas en inglés, *Ultra Narrow Band*). UNB ofrece una utilización eficiente del espectro que da como resultado un aumento en la capacidad de la red y un bajo consumo de energía. Admite velocidades de datos muy bajas en comparación con otras tecnologías LPWA ya que permite solo 140 mensajes de 12 bytes por día y cada transmisión tarda tres segundos. Para proporcionar confiabilidad, SigFox transmite el mensaje o paquete varias veces, dando como resultado un alto consumo de energía.

Opera en una Banda ISM. En Europa a 868 MHz y en Norteamérica a 915 MHz. Utiliza modulación por desplazamiento de fase binaria (BPSK-*Binary Phase Shift Keying*) en la banda ultra estrecha (100 Hz) la cual proporciona un nivel de ruido bajo. Sigfox ofrece un bajo consumo de energía y alta sensibilidad del receptor. Es decir, una mayor cobertura de área con un diseño simple de antena del dispositivo final. El diseño asegura una mayor vida útil de la batería, pero con un considerable costo de rendimiento. La tasa de datos de Sigfox es de solo 100 bps.

La transmisión funciona de la siguiente manera: un dispositivo transmite el mensaje tres veces en diferente frecuencia y periodo, asegurando razonablemente la tasa de entrega de mensajes

al 95%. El tamaño de la carga útil es de 12 bytes con 14 bytes de sobrecarga. La ventaja más significativa de Sigfox es que los operadores de todos los países solo despliegan y administran las redes de radio donde el OSS / BSS y la plataforma están centralizados y compartidos por todos los operadores del mundo, lo que permite una eficiente comunicación.

Comparación entre redes LPWAN

Existen escenarios de aplicación de IoT en los cuales las tecnologías LPWAN ofrecen una excelente solución, independientemente de cuál de ellas se elija. Algunos de estos escenarios son:

- Ciudades inteligentes. El objetivo de la ciudad inteligente es utilizar de manera eficiente los recursos públicos, mejorar la calidad de vida y reducir el costo de administración. Un ejemplo de aplicaciones de ciudades inteligentes es la gestión de residuos. Algunas ciudades inteligentes usan contenedores de basura igualmente inteligentes, que detectan el nivel de basura en su interior y envían la información a un centro de control que luego optimiza la ruta del camión recolector, reduciendo eventualmente el costo operativo. Otra aplicación es el alumbrado público. La iluminación inteligente reduce significativamente el costo de luz en las calles cambiando la intensidad de la luz según el entorno. También reduce el costo de mantenimiento al proporcionar monitoreo de fallas en tiempo real.
- Transporte y logística. La mayoría de los vehículos más nuevos incluyen sensores, capacidad de conexión en red y procesador. IoT puede utilizarlos para mejorar la experiencia de conducción de varias formas, como mejorar el uso compartido de la carretera, reporte de accidentes y detección de estacionamiento. Se requieren comunicaciones de largo alcance, bajo consumo de energía y bajo costo.
- Agricultura inteligente. El sector agrícola es uno de los más adecuados para adoptar IoT, pues se pueden medir

diversas variables físicas como: nutrientes del suelo, fertilizantes usados, semillas plantadas, humedad del suelo y temperatura de los productos almacenados a través de sensores.

- Aplicaciones sanitarias. El sector de la salud es un gran mercado para las aplicaciones de IoT. Ejemplos de aplicaciones de IoT para la atención médica incluyen monitoreo remoto de signos vitales (ej. presión sanguínea, temperatura), cuidado de ancianos, monitoreo de enfermedades crónicas, etc. Los requisitos clave para IoT en la mayoría de las aplicaciones relacionadas con la salud es que no sean invasivos y que establezcan una comunicación segura y confiable.

Como se describe en las secciones anteriores, la tabla 2 muestra una comparativa entre tres redes LPWAN con sus principales características.

Tabla 2. Comparación de características de LoRaWAN, NB-IoT y Sigfox

	LoRaWAN	NB-IoT	Sigfox
Frecuencia (MHz)	868 y 915	868 y 915	868
Modulación	FSS/CSS	OFDMA	D-BPSK
Descarga útil (bytes)	14	125	8
Carga útil (bytes)	51	125	12
Velocidad de datos (bps)	50 K	200 K	100
Latencia (seg)	Hasta 128	1.6-10	3
Duración de batería (años)	15	10	20
Alcance (km) (urbano y rural)	5 y 20	1 y 10	10 y 40

Fuente: Elaboración propia.

Conclusiones

Existen actualmente en el mercado muchas y variadas opciones para la implementación de escenarios de IoT, lo cual puede causar confusión sobre cómo conectar los dispositivos y qué tecnología utilizar. La recomendación básica para proceder con la implementación de una solución IoT es primero analizar el entorno en que se implementará; si es en un espacio pequeño podemos elegir entre las tecnologías de corto alcance como Bluetooth, NFC, Z-Wave, etc., en cambio, si el espacio de implementación es amplio, tendremos que elegir una tecnología LPWAN como LoRaWan, SigFox o NB-IoT. Posteriormente, la elección de una u otra tecnología tendrá que ver con el costo, la velocidad, la eficiencia, la exposición a interferencias y el consumo de energía, principalmente.

También es posible que los requerimientos del escenario de implementación de IoT no se satisfagan con las características de una sola tecnología, en cuyo caso se tendría que considerar una combinación de algunas de ellas, lo cual no es una idea descabellada puesto que puede encontrarse la manera de crear entornos donde convivan varias tecnologías diferentes.

En otros ámbitos de la tecnología como LANs, sistemas operativos, puertos serie, etc., diferentes productos y tecnologías convergen en un estándar o se han hecho compatibles, por ejemplo, ahora puede instalarse Windows en una Mac, Ethernet superó a Token Ring, USB se convirtió en un puerto básico para la conexión de dispositivos de E/S, entre otros. De igual manera, es posible que en el futuro las tecnologías para IoT existentes en la actualidad converjan en una sola, o terminen siendo tan compatibles que los usuarios finales o desarrolladores no tendrán que preocuparse de las características de la tecnología subyacente de los dispositivos sino simplemente de su funcionalidad.

Referencias

Baraza, A. de J. C. (2019). *Modelo de Implementación de Ciberseguridad para Sistemas IoT en el Marco de Redes 5G.* http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/12043/T005.8%20C393.pdf?sequence=1&isAllowed=y

Bluetooth SIG. (2021). Learn about Bluetooth. *Bluetooth® Technology Website*. https://www.bluetooth.com/learn-about-bluetooth/

Celer, V. (2020). *Un truco para facilitar el reclamo de equipaje en el aeropuerto*. https://www.celersms.com/baggage-claim-es.htm

Connectivity Standards Alliance. (2021). *Zigbee*. https://zigbeealliance.org/solution/zigbee/

Ismail, D., Rahman, M. y Saifullah, A. (2018). Low-power wide-area networks: Opportunities, challenges, and directions. En *19th International Conference on Distributed Computing and Networking* (pp. 1-6) https://doi.org/10.1145/3170521.3170529

Develco Products. (2021). *White label products for IoT solutions*. https://www.develcoproducts.com/products/

Hossain, M.I. y Markendahl, J.I. (2021). Comparison of LPWAN technologies: cost structure and scalability. *Wireless Personal Communications*, *121*, pp. 887-903. https://doi.org/10.1007/s11277-021-08664-0

IEEE 802.11 WLAN Working Group. (2021). *IEEE 802.11, The working group setting the standards for wireless LANs*. https://www.ieee802.org/11/

NFC Forum. (2021). *About the technology*. https://nfc-forum.org/what-is-nfc/about-the-technology/

Palattella, M. R., Dohler, M., Grieco, A., Rizzo, G., Torsner, J., Engel, T. y Ladid, L. (2016). Internet of things in the 5G era: enablers, architecture, and business models. *IEEE Journal on Selected Areas in Communications*, *34*(3), pp. 510-527. https://doi.org/10.1109/JSAC.2016.2525418

Ramírez, I. (2021). Qué es Wi-Fi 6 y qué ventajas tiene con respecto a la versión anterior. Xataka.com https://www.xataka.com/basics/que-wi-fi-6-que-ventajas-tiene-respecto-a-version-anterior

Smart Hubs. (2021). *Z-Wave*. https://www.z-wave.com/shop-z-wave-smart-home-products

Swedberg, C. (2021). *IoT, BLE enable emergency response to cardiac arrests. RFID Journal*. https://www.rfidjournal.com/iot-ble-enable-emergency-response-to-cardiac-arrests

Venco Electrónica. (2020, 3 de diciembre). *Qué es ZigBee, cómo funciona y características principales*. https://www.vencoel.com/que-es-zigbee-como-funciona-y-caracteristicas-principales/

Wi-Fi Alliance. (2021a). *Wi-Fi Alliance*. https://www.wi-fi.org/
Wi-Fi Alliance. (2021c). *Wi-Fi Aware | Wi-Fi Alliance*. https://www.wi-fi.org/discover-wi-fi/wi-fi-aware
Wi-Fi Alliance. (2021b). *Wi-Fi Certified 6 | Wi-Fi Alliance*. https://www.wi-fi.org/discover-wi-fi/wi-fi-certified-6
Zhiwei, X. y Ni, J. (2021). A study on key LPWAN technologies. *Journal of Physics Conference Series 1871*(1), 012011. https://doi.org/10.1088/1742-6596/1871/1/012011
Z.Wave Alliance. (2021). *Learn*. https://www.z-wave.com/learn

Capítulo 5

Protocolos de comunicación de los dispositivos IoT

Omar Álvarez Cárdenas
Margarita Glenda Mayoral Baldivia
Raúl T. Aquino Santos

Resumen

Internet de las cosas (IoT) se está convirtiendo en un área de oportunidad para aprovecharse en todas las áreas del conocimiento e incluso para situaciones de la vida cotidiana. Para comprender mejor su funcionamiento, capacidades y aplicación, es indispensable hacer un análisis muy particular desde el punto de vista de aplicaciones y protocolos involucrados en el tema. En los siguientes apartados se hace referencia a los diversos protocolos de comunicaciones que permiten, en cada una de las arquitecturas por capas, definir la manera en la cual es posible interconectar múltiples dispositivos de bajo costo, poco ancho de banda y bajo consumo energético, con o sin la pila de protocolos TCP/IP. Para describir las funciones de cada protocolo de comunicaciones, se hace referencia a la pila de seis capas para IoT propuesto por la Internet Engineering Task Force (IETF).

Introducción

Para conectar las cosas a Internet no basta con poner direcciones IPs a los dispositivos a conectar, esto es solo el primer paso para la realización del Internet de las cosas (IoT); es decir, las redes IP hacen posible el intercambio de información entre las cosas que generalmente se conectan a una red no-IP, mientras que para las redes IP las distancias no son problema, las redes no-IP son redes más simples con limitaciones en el alcance y, por lo tanto, se requieren protocolos de comunicación para IoT.

Los protocolos IoT son un conjunto de normas, reglas y algoritmos que permiten el intercambio de información entre dos entidades de una red, facilitando la comunicación Máquina a Máquina (M2M). Un protocolo de comunicación de dispositivos IoT es el equivalente al idioma, el lenguaje corporal, mientras que el contexto que utilizamos los humanos para comunicarnos, entendida como la forma de comunicación entre dos dispositivos, estará definida por el protocolo y las características de comunicación que dicho protocolo establezca.

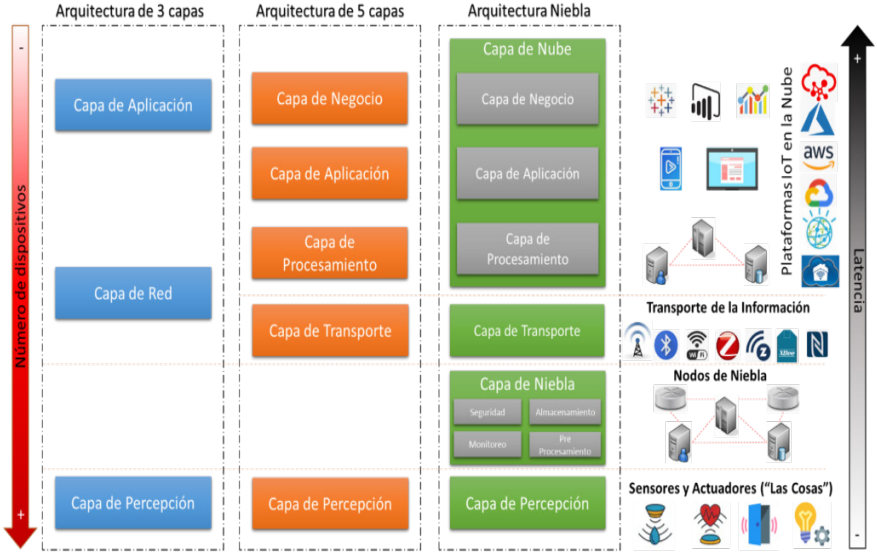
Los protocolos de comunicación para los dispositivos IoT han sido desarrollados con características que permiten transmitir cualquier tipo de información, con un consumo mínimo de recursos, esto los hace altamente flexibles, atendiendo los principios de una red de comunicación IoT, tales como bajas velocidades de datos, bajo consumo de energía, conectividad bidireccional segura, largo alcance en la comunicación y, en ocasiones, favorecer la baja latencia.

La conexión de los dispositivos IoT generalmente se implementa a través de la pila de protocolos TCP/IP, esta pila de protocolos es compleja y necesita gran cantidad de memoria y energía, por tal razón ha sido modificada para que los dispositivos IoT puedan conectarse local y generalmente a través de redes no IP, ya que en estas redes el consumo de energía es menor y la conexión a internet se realiza a través de una pasarela (*gateway*).

En el modelo de referencia OSI encontramos una guía de los protocolos a utilizar tanto en los niveles físico, nivel de vinculación o de enlace de datos, de red, de transporte, así como el nivel de aplicación. El tipo de protocolo de comunicación a utilizar

dependerá de las características del sistema de red en el que se transmitirán los datos y los componentes de la arquitectura IoT que puede estructurarse en tres capas, cinco capas, o bien en la arquitectura Niebla, según se muestra en la figura 1.

Figura 1. Arquitecturas IoT Relevantes



Fuente: Elaboración propia, basada en García (2020).

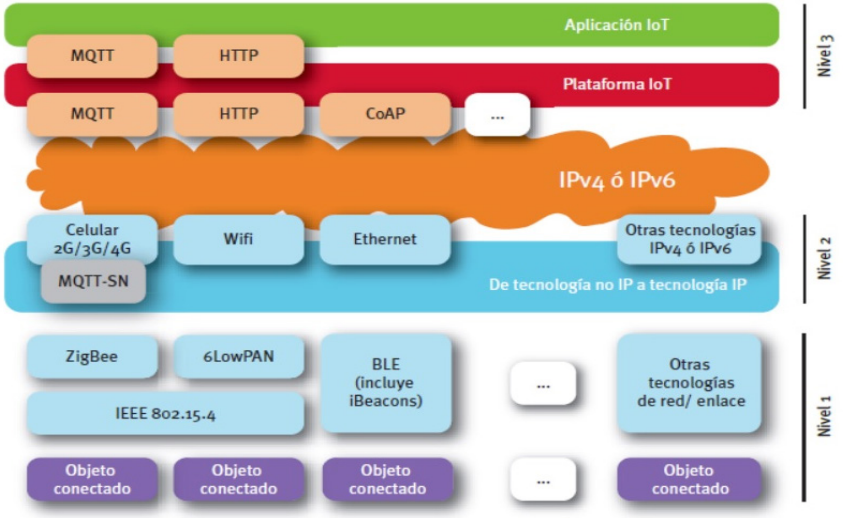
En general existe un gran número de arquitecturas y varias pueden ser la solución a un mismo problema, esto quiere decir que en un proyecto específico o aplicación a implementar en el IoT se podrán diferenciar tipos fundamentales de arquitecturas, por ejemplo:

- Arquitectura de dos niveles con objetos conectados sin protocolo IP.
- Arquitectura de dos niveles con objetos conectados con protocolo IP.
- Arquitectura de tres niveles con objetos conectados sin protocolo IP.

La arquitectura más comúnmente utilizada para conectar un gran número de dispositivos de bajo costo, baja capacidad y bajo

consumo de energía es la de tres niveles sin protocolo IP (figura 2), utilizando un *gateway* o repetidor para salir a una red IP.

Figura 2. Arquitectura de tres niveles - objetos conectados sin protocolo IP



Fuente: Elaboración propia, basada en Pardal-Garcés (2017).

En el Internet de las cosas la variedad de objetos a interconectar es enorme y estos pueden ser de tamaños muy pequeños, como sensores, actuadores, etc., o grandes como servidores, donde se recolectarán, serán procesados y se almacenarán los datos. Debido a esto, el elegir cuál de los protocolos conviene utilizar en un proyecto IoT puede ser complicado y se hace teniendo las consideraciones siguientes:

- Que sea escalable, es decir, que permita poner o quitar dispositivos sin afectar el comportamiento general de la comunicación en la red.
- Que no haya codependencia en que deba existir un dispositivo determinado para poder funcionar.
- La interoperabilidad es importante para permitir la funcionalidad entre dispositivos con distintos sistemas

operativos, desarrollos en distintos lenguajes de programación, etc.
- Que soporte comunicaciones simultáneas, con respuestas rápidas y tiempos de procesamiento variables acordes a las necesidades del sistema IoT.
- Brindar acceso fácil a los dispositivos e, indudablemente, proporcionar especificaciones de seguridad.
- Es importante también considerar diversos factores relacionados con la aplicación, las distancias entre dispositivos, la energía que se utiliza, la frecuencia de comunicación, entre otros.

Algunos protocolos de la capa física se abordan en el capítulo 4, mientras que en el presente capítulo se mencionan algunas notas referentes a los protocolos del nivel físico y se describen los protocolos del nivel de vínculo o enlace de datos en adelante.

*Nivel f*ísico

En el nivel físico del IoT se define el canal de comunicación entre los dispositivos de un entorno o sistema IoT particular o específico. Los protocolos IoT de la capa física ya han sido abordados en el capítulo anterior, estos protocolos son considerados como protocolos de acceso de red, los cuales funcionan dependiendo del tipo de red ya sean de corto alcance y bajo consumo, o redes de área extensa de bajo consumo de energía.

Los protocolos de redes de corto alcance son: Bluetooth *Low Energy* (BLE), *Near Field Communication* (NFC), Wi-Fi/802.11, Z-Wave, Zigbee; y Ethernet, mientras que los protocolos de redes de área extensa de bajo consumo son: 4G LTE para IoT, 5G para IoT, Cat-0, Cat-1, LoRaWAN, LTE Cat-M1, Banda estrecha o NB-IoT/CAT-M2, Sigfox.

Protocolo Bluetooth *Low Energy* (BLE)

La evolución de la tecnología Bluetooth clásica a Bluetooth de baja energía conocida también como Bluetooth Smart y una versión posterior llamada Bluetooth Smart Ready, definieron a esta tecno-

logía como viable en el IoT. Sin embargo, un punto a considerar al utilizar esta tecnología es la retrocompatibilidad entre dispositivos, dicha retrocompatibilidad según se muestra en la Figura 3, solo puede encontrarse en Bluetooth *Smart Ready*.

Figura 3. Compatibilidad de las versiones de Bluetooth

Fuente: Recuperada de https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works.

BLE no está diseñado para transferencia de grandes volúmenes de datos y es más adecuado para paquetes pequeños. Si se necesita mayor demanda de datos, conviene el Bluetooth Classic. Se integra fácilmente con smartphones y muchos dispositivos móviles.

Las principales características del protocolo son:

- Frecuencia: 2.4 GHz (ISM).
- Alcance: 50-150 m Y, a partir del Bluetooth 5.0, hasta 240 metros.
- Tasas de datos: 1 Mbps, alcanzando velocidades de hasta 50 Mbps.

Long Term Evolution (LTE)

LTE o Evolución a largo plazo es una tecnología de transmisión de datos diseñada para mejorar las velocidades de banda ancha inalámbrica para la creciente demanda del acceso de telefonía móvil (teléfonos 4G) y de dispositivos portátiles a Internet. 4G utiliza LTE

Advanced, técnicamente LTE o 4G LTE hacen referencia a un estándar de comunicación móvil llamado conectividad LTE desarrollado por la organización 3GGP (3rd *Generation Partnership Project*).
La ventaja principal entre LTE y LTE Advanced radica en la velocidad de navegación, y se dice que LTE Advanced se caracteriza por alcanzar velocidades de hasta 950 Mbps, razón por la cual se le identifica como 4G. Cabe señalar que mientras que LTE puede alcanzar velocidades de 1 a 170 Mbps y hasta 450 Mbps (en función de las características de las antenas), el 4G alcanza velocidades de 1Gbps; por esta razón, solo LTE Advanced es considerada 4G (Moes, 2020).

Protocolo Near Field Communication (NFC)

NFC, comunicación de campo cercano o la transmisión de datos en proximidad, es una tecnología de comunicación inalámbrica de apenas unos 10 o 15 centímetros de distancia; es decir, de muy corto alcance y frecuencia alta, opera en la frecuencia 13.56 MHz, por lo tanto, no se requiere licencia para operar, la transmisión de los datos se lleva a cabo en una interconexión entre dispositivos de una manera intuitiva, sencilla y simple.

La tecnología NFC es una extensión del estándar ISO/IEC-14443 para tarjetas de proximidad sin contactos, que combina la interfase de una tarjeta inteligente y un lector en un único dispositivo, lo que lo hace compatible con toda la infraestructura de pago sin contactos y de transporte existente actualmente.

NFC es una tecnología complementaria pues no se diseñó para la transmisión masiva de datos, por lo tanto, no sustituye el uso de otras tecnologías como Wi-Fi o Bluetooth.

La comunicación entre dispositivos NFC se realiza a través de un diálogo entre un dispositivo denominado *iniciador* y uno o varios dispositivos denominados *destino*, debiendo responder estos antes de recibir otra petición.

Existen dos modos de establecer la comunicación: *activo y pasivo*. En modo pasivo, el dispositivo *iniciador* genera el campo electromagnético y el dispositivo *destino* se comunica con este modulando la señal recibida. En este modo, el dispositivo *destino* obtiene la energía necesaria para funcionar del campo electromagnético generado por el *iniciador*.

En modo activo, tanto el dispositivo *iniciador* como el *destino* se comunican generando su propio campo electromagnético. De este modo, ambos dispositivos requieren de una fuente de alimentación para funcionar. Cuando el dispositivo funciona en modo pasivo, el receptor solo se utiliza para establecer la comunicación y confirmar la recepción de los datos. Sin embargo, en modo activo se requiere que ambos nodos negocien el intercambio de datos. Aunque muchas aplicaciones requieren que los dispositivos involucrados sean activos, la combinación de uso activo/pasivo puede ser útil para comunicarse con elementos sin batería, como pueden ser las tarjetas sin contactos o las etiquetas RFID que no dispongan de fuente de alimentación propia.

Power Line Communication (PLC)

El término comunicaciones por línea eléctrica (PLC) se refiere a la transmisión de señales de datos a través de la infraestructura de distribución de electricidad existente de bajo, medio y alto voltaje (LV, MV, HV).

PCL es una tecnología que inicialmente funcionaba en banda estrecha y ha evolucionado a redes de acceso de banda ancha. Se ha venido desarrollando con el respaldo de la ITU-T e IEEE, por mencionar algunas especificaciones en los estándares ITU-T G.hn y en el grupo de trabajo IEEE1901.

IEEE P1901/banda ancha sobre líneas eléctricas: IEEE P1901 es un estándar de banda ancha sobre líneas eléctricas (BPL) que se completó recientemente. BPL tiene altas velocidades de datos superiores a 100 Mbps utilizando frecuencias por debajo de 100 MHz. Aunque puede operar alrededor de 30 MHz y lograr velocidades de datos más altas, la alta atenuación a bajas frecuencias limita el rango de comunicación; puede admitir más de 2,000 dispositivos por lo que no es la preferida (Hussein y Melike, 2015).

ITU-T G.hn: ITU-T desarrolló el estándar G.hn para la comunicación en locales residenciales, oficinas de pequeña escala, hoteles y salas de conferencias. G.hn puede operar sobre todo tipo de cableado doméstico, incluida la línea telefónica, la línea eléctrica, el cable coaxial y el cable Cat-5. G.hn, puede admitir tasas de bits de hasta 1 Gbps y hasta 250 nodos. Los dispositivos G.hn pre-

tenden ser interoperables con dispositivos de línea eléctrica que utilicen el estándar IEEE P1901.

Actualmente, se pueden utilizar convertidores PCL a Wi-Fi para generar sistemas híbridos de redes domésticas principalmente para usos de control, telemetría, monitoreo, distribución de señales de Línea de Suscriptor Digital (DSL), transmisión de audio, de televisión y de cámaras web.

Modulación utilizada en PLC

PLC utiliza como sistemas de modulación:

- Multiplexación por división de frecuencia ortogonal (OFDM).
- Modulación por desplazamiento de fase binaria (BPSK)..
- Modulación por desplazamiento de frecuencia (FSK y S-FSK).
- Modulación por desplazamiento de código diferencial (DCSK).

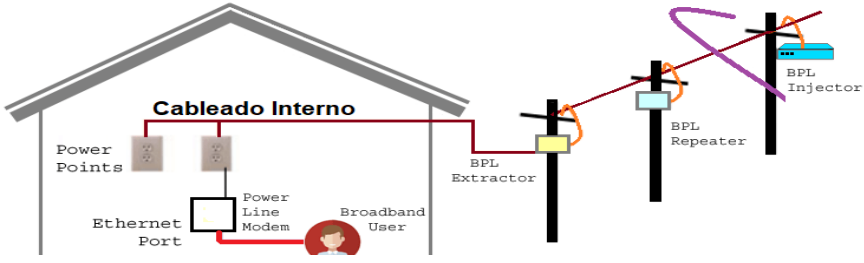
Tipos de PCL

Según el ancho de banda las tecnologías PLC se clasifican en tecnologías:

- PLC de Banda Ultra Estrecha (UNB-PLC): Transmisión de datos en frecuencias inferiores a 3 KHz, con velocidades muy bajas (cientos de bps); esta tecnología puede alcanzar largas distancias ya que se ven menos afectadas por las pérdidas de transmisión.
- PLC de Banda Estrecha (NB-PLC): La transmisión de estos sistemas se realiza en frecuencias entre los 3 y 500 KHz. Esta característica permite dividir esta tecnología en tecnologías de baja velocidad de datos (LDR) y alta velocidad de datos (HDR).
- Las tecnologías LDR utilizan modulación de una sola portadora y transmiten velocidades de datos de pocos kbps; mientras que las HDR se basan en modulaciones multiportadoras y transmiten velocidades de cientos de kbps hasta 1 Mbps.

- PLC de Banda Ancha (BPL): Opera en frecuencias desde 1 hasta 250 MHz, permite un ancho de banda de 45 Mbps y alcanza velocidades de hasta 200 Mbps. En la figura 4 se ilustran los tipos de PLC.

Figura 4. Sistemas PCL



Fuente: Elaboración propia, basado en Abhimanyu, (2019).

PLC actualmente se utiliza como red eléctrica inteligente en las denominadas *Smart Grid* que combinan las TIC con la automatización y el control, la cual es considerada como tecnología sostenible.

Nivel vínculo de datos

Los protocolos de vínculo de datos son los encargados de transportar los datos en el sistema IoT, deben permitir la comunicación entre los dispositivos de forma simultánea, en esta capa se pueden corregir errores de la capa física. Los protocolos de transmisión se utilizan para codificar la información.

Estándar IEEE 802.15.4

Es un estándar abierto que define el nivel físico y de control de acceso al medio de redes inalámbricas de área personal (WPAN), opera en las bandas libres de 2.4 GHz, para Estados Unidos la 915 Mhz, mientras que para Europa, opera en 868MHz. 868/915 MHz: 11 canales. 2.4 GHz: 16 canales.

Es uno de los estándares principales para la implementación de IoT, que permite generar entornos perfectos siempre que haya una baja relación señal a ruido. Utiliza CSMA/CA para gestión del medio y prevención de colisiones y es posible utilizar ranuras de tiempo TDMA (Acceso Múltiple por División de Tiempos) para aplicaciones de baja latencia. Es un estándar que establece las capas PHY y MAC de muchos protocolos de capas superiores como por ejemplo Zigbee.

Zigbee es el nombre de un conjunto de protocolos de alto nivel, basado en el estándar 802.15.4. Esta tecnología utiliza las bandas de frecuencias ISM para usos industriales, científicos y médicos.

Tipos de dispositivos y nodos o modos de operación definidos por el estándar

El estándar define dos tipos de topologías: de estrella y de punto - punto, esta última puede implementarse en estructura de malla (*mesh*) o en grupos de árboles (*cluster tree*).

Una red Zigbee puede instalarse en topología estrella, malla o en conjuntos de árboles (*cluster tree*), el número de nodos en la red dependerá de cómo estén organizados, es decir, pueden tenerse redes de 254 nodos hasta 64,770 siempre que esta cantidad se agrupe en 255 clústeres de nodos, cada clúster con 254 nodos.
IEEE 802.15.4 define dos *tipos de dispositivos*:

- *Full Function Device* (FFD): Dispositivos de Función Completa, pueden funcionar en cualquier topología, sea estrella, malla o grupos de árboles, reciben el nombre de coordinador o coordinador de red.
- *Reduced Function Device* (RFD): Dispositivo que puede ser miembro de una red estrella, solo se comunica con el coordinador de la red y se trata de dispositivos de baja complejidad, poco requerimiento de memoria y de procesamiento.

Se requiere que en las redes Zigbee al menos un dispositivo sea FFD y que sea el coordinador de la red y todos los nodos finales de la red puedan ser RFD.

En cuanto a los nodos el estándar, define tres *tipos de nodos:*

- Coordinador: es indispensable la existencia de un nodo coordinador en la red Zigbee; en una topología de árbol será el nodo raíz, se trata de un dispositivo FFD que es responsable del arranque de la red, de admitir nodos, asignar direcciones de red, debe tener fuente de alimentación permanente y segura, en este nodo se requiere más potencia de cómputo.
- Enrutador: se trata de un dispositivo FFD que no es coordinador y se utiliza para extender la cobertura de la red y proporcionar confiabilidad en la creación de rutas principales y adicionales en la red.
- Dispositivo final: se trata de dispositivos RFD que tienen menos potencia de cómputo, se alimentan con baterías, se comunican con un nodo coordinador o con un enrutador.

Otros protocolos que tienen como base el IEEE802.15.4 son: 6LoWPAN, WirelessHART, Z-Wave, Bluetooth y ULP Bluetooth.

Lower Power Wide Area Network (LPWAN)

Las LPWAN tal como su nombre lo dice, es el nombre genérico que hace referencia a las tecnologías de redes inalámbricas de área extensa en la que se utiliza baja potencia para la transmisión de los datos, en la que la transmisión se realiza entre un dispositivo y una estación base/*gateway* que pueden estar a centenares de metros o kilómetros. Estas tecnologías trabajan en las bandas de radiofrecuencia ISM (siglas en inglés Industrial, Científica y Médica) dicha banda representa un espectro reservado internacionalmente al uso no comercial y se asocia a la industria, la ciencia y los servicios médicos.

Los factores más críticos en una LPWAN son:

- La arquitectura de la red.
- Rango de comunicación, es decir, los alcances.
- Duración de la batería o bajo consumo.
- Robustez a la interferencia.
- Capacidad de la red (número máximo de nodos en una red).

- Seguridad de la red.
- Comunicación unidireccional frente a bidireccional.
- Variedad de aplicaciones servidas.

Las principales características que hace de LPWAN como tecnología que satisfacen los requerimientos de IoT son las siguientes:

- Las distancias pueden ser kilómetros y no metros, superiores a los 800 kilómetros.
- El bajo consumo de energía, el uso de baterías de larga duración, pues en lugar de semanas o meses llegan a durar años.
- La cantidad de datos transmitidos no permite manejar grandes cantidades de datos por lo que transmisiones de fotos y videos quedan descartados.

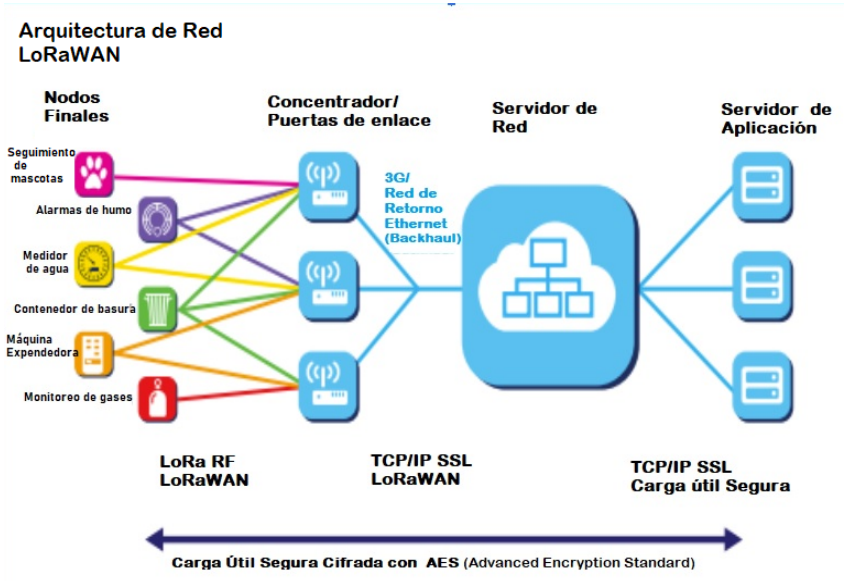
Son tecnologías que permiten instalar decenas o centenas de nodos distribuidos en una gran área geográfica, se trata de tecnologías diseñadas específicamente para entornos de IoT, como redes de sensores que transmiten datos periódicamente en paquetes pequeños de información. No son adecuadas para las aplicaciones comerciales de transmisión de voz, audio e incluso para algunos servicios de mensajería de textos por el limitado ancho de banda.

Algunos ejemplos de estas tecnologías que trabajan en el espectro no reglamentado son: SigFox, LoRaWAN, RPMA, mientras que NB-IoT y LTE-M ofrecen servicios en espectro concesionado o de uso exclusivo.

SigFox es una tecnología LPWAN propietaria, se considera una de las redes más grandes del mundo, utiliza una topología de estrella, que puede comunicar a distancias de hasta 50 Km. Utiliza bandas de frecuencias sin licencia (868 o 902 MHz). La capacidad de respuesta del dispositivo emisor era muy limitada por lo que no se consideraba apta para comunicación bidireccional. Actualmente, Sigfox admite comunicaciones bidireccionales, utiliza el ancho de banda tan eficientemente que ha experimentado niveles de ruidos muy bajos.

LoRaWAN Es un estándar abierto, un protocolo de comunicaciones basado en los *chips* LoRa que permite la conexión de dispositivos en redes regionales, nacionales o globales en una topología de estrella extendida, que permiten conexiones operando de forma bidireccional. Los dispositivos LoRa se conectan a una o varias puertas de enlace, las cuales realizan la transmisión entre los dispositivos finales y un servidor central, que se conectan a redes IP estándar en una conexión normal, en la figura 5 se muestra una arquitectura de red.

Figura 5. Arquitectura de una red LoRaWAN



Fuente: Elaboración propia, adaptado de Lora (2015).

Tipos de dispositivos LoRaWAN

No todos los dispositivos finales son iguales, para optimizar la red se utilizan perfiles de dispositivos, dependiendo de la aplicación y los diferentes requisitos de dichas aplicaciones. Las clases de dispositivos compensan la latencia de comunicación del enlace descendente de la red, frente a la vida útil de la batería. Por ejemplo:

en una aplicación de tipo control o actuador en el enlace descendente, la latencia de la comunicación es un factor importante. Se definen dispositivos Clase A, Clase B y Clase C.

Los dispositivos finales bidireccionales Clase A: permiten comunicaciones en las que la transmisión de enlace ascendente de cada dispositivo final es seguida por dos ventanas de recepción de enlace descendente corto. La ranura de transmisión programada por el dispositivo final se basa en sus propias necesidades de comunicación con una pequeña variación en una base de tiempo (tipo de protocolo ALOHA). En la operación, los dispositivos Clase A utilizan eficientemente la energía, este dispositivo final es apto para aplicaciones que solo requieren comunicación de enlace descendente desde el servidor, poco después de que el dispositivo final haya enviado una transmisión de enlace ascendente; es decir, el enlace descendente para las comunicaciones desde el servidor solo estará disponible después de que el sensor transmite y tendrá que esperar hasta el próximo enlace ascendente programado para recibir información.

Dispositivos finales bidireccionales Clase B: son dispositivos finales bidireccionales con ranuras de recepción programadas en los que, además de las ventanas de recepción aleatoria de Clase A, abren ventanas de recepción adicionales en horarios programados. Para que el dispositivo final abra su ventana de recepción en una hora programada, recibe una baliza sincronizada en el tiempo de la puerta de enlace, esto permite que el servidor conozca cuándo el dispositivo final está escuchando.

Dispositivos finales bidireccionales con ranuras de recepción máxima Clase C: los dispositivos finales de Clase C tienen ventanas de recepción abiertas casi continuamente, solo se cierran durante la transmisión. Estos dispositivos pueden darse el lujo de escuchar continuamente sin latencia para la comunicación del enlace descendente.

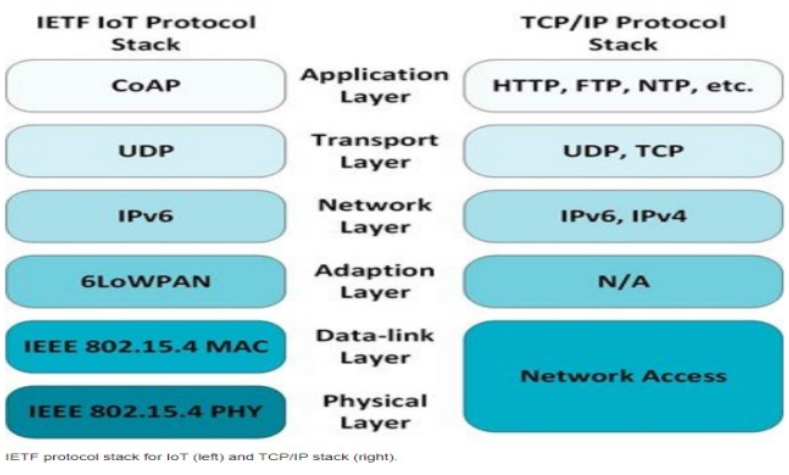
La seguridad de LoRaWAN se implementa utilizando dos capas de seguridad: una para la red y otra para la capa de aplicación. En la capa de seguridad de la red se garantiza la autenticidad del nodo de la red mientras que en la capa de aplicación garantiza que el operador de la red no tenga acceso a los datos de la aplicación

del usuario final; utiliza el cifrado AES (*Advanced Encryption Standard*) con el intercambio de claves utilizando un identificador IEEE EUI 64, este proceso utiliza la dirección MAC de Ethernet de 48 bits de un cliente e introduce otros 16 bits en medio de la dirección MAC de 48 bits, para crear una ID de interfaz de 64 bits.

Nivel de red

En el nivel de red del IoT permite la comunicación entre los dispositivos y el enrutador. En este nivel se utilizan los protocolos de red existentes que son necesarios en cualquier conexión a Internet, sea esta conexión desde el hogar, las oficinas, la industria, el campo, etc. Protocolos como IP y UDP de la suite TCP/IP son utilizados en el modelo de las pilas de protocolos de la IETF (Grupo de trabajo de ingeniería de Internet), tal como se muestra en la figura 6, siendo IP el protocolo de este nivel.

Figura 6. Comparativa de la Suite TCP/IP y la pila de protocolos IoT



Fuente: Elaboración propia, adaptado de Tameem (2021).

IP

El IP es fundamental para la IoT. Actualmente existen IPv4 e IPv6, ambos sirven para identificar máquinas conectadas a una red, el IP permite la interoperabilidad de los sistemas. El direccionamiento de dispositivos en la red no es un tema menor, aunque, tal como se muestra en la figura 6, se considera el IPv6 como el más adecuado y es el único definido en la pila de protocolos IoT (IETF). Cabe señalar que Ethernet/Wi-Fi y 6LoWPAN dependen en gran medida tanto de IPv4 como de IPv6.

La característica principal que muestra claramente por qué IPv6 es el más adecuado para el IoT es la cantidad de dispositivos que se pueden direccionar con cada versión de IP. Mientras que el IPv4 permite direccionar un total de 4,294,967,296 (2^{32}) direcciones diferentes con los 32 bits que conforman dicha dirección, las cuales resultan insuficientes para el propósito del IoT, IPv6 utiliza 128 bits; por lo tanto, la cantidad de direcciones disponibles en esta versión es de 340,282,366,920,938,463,463,374,607,431,768,211,456 (2^{128} o 340 sextillones) direcciones por cada milímetro cuadrado de la superficie de la Tierra, con lo que sobra decir que son direcciones suficientes para el IoT.

En cuanto a los avances respecto al uso del IP en el IoT y el desafío de la interoperabilidad de las tecnologías, debido a la gran cantidad de estándares existentes, el IETF (*Internet Engineering Task Force*) ha publicado un nuevo estándar clave para IoT. Esta tecnología, llamada SCHC, permite el uso de protocolos basados en IP en redes IoT. Garantiza la interoperabilidad entre las tecnologías LoRaWAN, Sigfox, NB-IoT y LTE-M. El nuevo estándar internacional, llamado SCHC (pronunciado "chic") para la compresión de encabezado de contexto estático, es un mecanismo avanzado para fragmentar y comprimir encabezados de pila de Internet. Permite que dispositivos restringidos conectados a redes LPWAN se comuniquen a través de IP.

Nivel de transporte

La suite de protocolos TCP/IP en el nivel de transporte define el uso de protocolos TCP o UDP. En cualquier protocolo de IoT, el nivel de transporte habilita y protege la comunicación de los datos mientras viajan entre niveles. La pila de protocolos referenciada en la figura 6 como la pila de protocolos IoT de la IETF, muestra que para el nivel de transporte del IoT se considera el protocolo UDP.

UDP

El protocolo de datagramas de usuario (UDP) es el protocolo de capa de transporte más simple, que se utiliza principalmente para establecer conexiones de baja latencia y tolerancia a pérdidas entre aplicaciones en la red de comunicación. Tanto TCP como UDP se ejecutan sobre el Protocolo de Internet (IP), por lo que se denomina TCP/IP y UDP/IP.

UDP es un protocolo sin conexión, lo que significa que el remitente simplemente transmite los datos, sin esperar la conexión con el receptor. Es un protocolo poco fiable, en comparación con TCP. No existe un mecanismo de verificación de errores o un mecanismo de corrección involucrado en la transmisión de datos que resulte en el uso de menos ancho de banda. El protocolo UDP solo envía los paquetes (o datagramas). No hay garantía de acuse de recibo del paquete recibido por el otro extremo. Permite menos sobrecarga de datos y retrasos.

Para lograr un mayor rendimiento, el protocolo permite que los paquetes individuales se descarten (sin reintentos) y que los paquetes UDP se reciban en un orden diferente al que fueron enviados, según lo dicte la aplicación.

El protocolo UDP proporciona muy pocos servicios de recuperación de errores, ofreciendo en su lugar una manera directa y rápida de enviar y recibir datagramas a través de una red; es empleado fundamentalmente para la comunicación entre sistemas, en los que la velocidad es más importante que la fiabilidad en la transmisión de la información.

Se utiliza UDP para enviar paquetes de datos desde un emisor a un receptor, sin importar que este último los reciba todos

completamente. Cada paquete es enviado directa e individualmente sin establecerse ni reconocerse un canal de datos fiable.

En términos de la velocidad de envío, no existe un orden inherente en la transmisión de los paquetes de datos, pues todos se envían a través de la red de forma independiente entre sí. De igual modo, no es posible solicitar los paquetes de datos que faltan una vez que se pierden en tránsito.

Nivel de aplicación

El sistema IoT se conforma de dispositivos, datos, conectividad y usuarios tecnológicos, estos usuarios interactúan con las tecnologías y los dispositivos IoT, en el uso, generación, administración e implementación de los datos de tal manera que el nivel de aplicación tiene la responsabilidad de proporcionar una interfaz que permita la comunicación entre usuarios y dispositivos a través de un protocolo IoT determinado.

Los protocolos IoT para el nivel de aplicación más comunes son el AMQP (*Advanced Message Queuing Protocol*), CoAP (*Constrained Application Protocol*), DDS (*Data Distribution Service*) y MQTT (*Message Queue Telemetry Transport*). La finalidad de dichos protocolos es proveer los mecanismos para que los dispositivos IoT tengan la capacidad de publicar sus datos a través de la red de Internet, utilizando algunas de las soluciones de envío de mensajes en la capa de aplicación. La adecuada selección de estos protocolos en la capa de aplicación para garantizar el intercambio de información de manera estándar y efectiva no es una tarea fácil, por ello se deben conocer a detalle sus características y funcionamiento de acuerdo con las necesidades de nuestro sistema IoT.

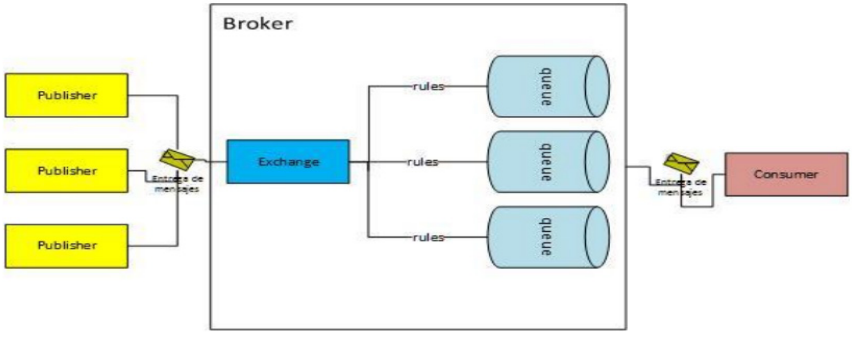
Advanced Message Queuing Protocol (AMQP)

El AMQP fue desarrollado en Inglaterra en el año 2003 por John O'Hara, está definido para conexiones M2M como un protocolo abierto de mensajería ligero en ambientes que requieren fiabilidad, seguridad e interoperabilidad. El intercambio de mensajes se realiza a través de la capa de transporte utilizando el puerto 5674 del protocolo TCP y su arquitectura es cliente/servidor (Caiza et

al., 2020). Mediante esta forma de trabajo, AMQP asegura que la información que envía sea de una manera segura y eficiente entre las aplicaciones en entornos basados en nubes privadas o públicas, así como dentro de infraestructuras móviles. Este protocolo tiene un gran potencial para la integración de programas y reducción de costos, debido a que evita el uso de tecnologías propietarias gracias a su filosofía de interoperabilidad abierta.

Para establecer la comunicación, AMQP implementa en su arquitectura cliente/servidor elementos denominados *brokers* de mensajería, que se utilizan para la transmisión de los mensajes. Estos, además de ser regulados por el protocolo AMQP, están constituidos por dos componentes principales como se aprecia en la figura 7. El primero de ellos es el intercambiador (*exchanges*) cuya finalidad es enrutar los mensajes al segundo elemento del *broker* que son las colas (*queue*), adecuadas mediante la aplicación de ciertas reglas y condiciones.

Figura 7. Arquitectura AMQP



Fuente: Elaboración propia, adaptada de González y López (2017)

Para asignar adecuadamente un mensaje se manejan cuatro intercambiadores:

- Intercambiador directo: hace el envío de mensajes a un receptor definido aplicando claves de enrutamiento y donde cada cola tiene una clave de vinculación, cuando estas coinciden, el mensaje se puede enviar a la cola y al receptor (consumidor) correspondiente. Se puede

presentar el caso que una cola tenga varias claves de vinculación que le permitan utilizarse para varias claves de enrutamiento. El caso opuesto sería que varias colas compartan una clave de vinculación, provocando que el intercambiador multiplique el mensaje y lo mande a distintos destinatarios.

- Intercambiador de abanico: es similar al anterior, pero aquí el *broker* ignora por completo la clave de enrutamiento y permite que el intercambiador envíe un mensaje a todas las colas existentes con una copia de la información.
- Intercambiador por tema definido: permite que las claves de vinculación y enrutamiento se emparejen utilizando comodines para que ambas claves no tengan una coincidencia exacta entre ellas, de esta manera, es posible hacer el envío de mensajes específicos para varias colas de manera simultánea.
- Intercambiador de encabezamientos: en lugar de utilizar la clave de enrutamiento, esta opción se enfoca a trabajar con el encabezamiento de un mensaje. Analiza el argumento de la etiqueta *x-match* para asegurar que todos los valores coincidan (valor:*todos*) para comportarse como un intercambiador directo. La otra forma es verificando que la etiqueta sea: "valor:*cualquiera*", para generar el mismo procedimiento descrito en el intercambiador por tema definido.

Las colas son el otro elemento que conforma el *broker*, su finalidad es el almacenamiento de la información, para ello debe identificarse con un nombre, el cual se asigna por el cliente o por el *broker*. Una cola es una memoria que puede integrarse en un disco duro para almacenarse de forma permanente o bien puede ser volátil en una memoria no permanente. La ventaja de ser una cola permanente radica en que la cola persiste, sin importar que se reinicie el *broker*. Estas funcionalidades permiten establecer dos maneras de recuperar un mensaje que se pierde en una cola. La primera faculta al cliente de requerir que se genere un acuse de re-

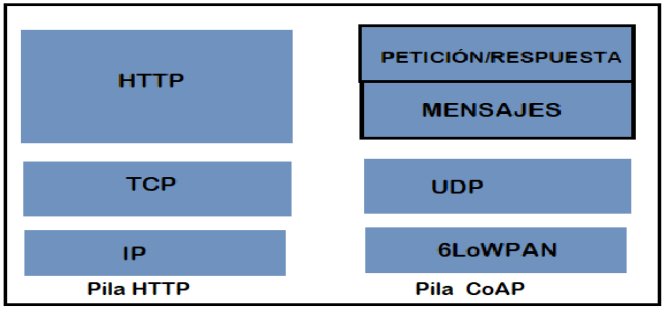
cibo y la segunda es no requerir de ese acuse. Si se elige la primera opción y el consumidor no envía ningún mensaje, el *broker* intentará enviar el mensaje a otro consumidor o tratará de conectar con el mismo receptor una vez más. Sin embargo, si está activada la variante sin confirmación y el consumidor no reclama la entrega del mensaje, este se perderá (IONOS, 2019).

El estándar AMQP nos ayuda a definir el comportamiento del servidor que maneja los mensajes hasta el cliente de mensajería, proporcionando la suficiente interoperabilidad sin importar que se apliquen implementaciones de diferentes proveedores, esto lo logra debido a que se comporta como un protocolo de nivel de cable (*wire level protocol*) que maneja el formato de los datos como un flujo de octetos. Esto genera que cualquier otra herramienta que cumpla con estas especificaciones pueda interoperar con cualquier herramienta que cumpla con este protocolo sin importar el lenguaje de programación con el cual se generó (ISO/IEC, 2014).

Constrained Application Protocol (CoAP)

El protocolo de aplicación restringida (CoAP) definido en el RFC 7252, es un protocolo de transferencia web especializado en el Internet de las cosas. Es un protocolo de aplicación similar al protocolo HTTP, ambos utilizan el modelo web REST. CoAP maneja intercambio de mensajes de forma asíncrona, ha sido diseñado para funcionar en dispositivos de recursos limitados, que no pueden ejecutar HTTP, tales como sensores, microcontroladores, etc.; es decir, en entornos limitados definidos en el estándar CoRE (*Constrained RESTful Environments*). En CoAP el intercambio de mensajes se realiza entre un cliente y un servidor sin necesidad de elementos intermedios en la comunicación, y se lleva a cabo a través de solicitudes y respuestas.

Figura 8. Pila de protocolo HTTP y pila de protocolo CoAp



Fuente: Elaboración propia, adaptado de Kowalewski (2018).

CoAP utiliza un enfoque de dos capas para funcionar. Una capa transaccional llamada *message* o mensajes que es utilizada con UDP y la naturaleza asincrónica de las interacciones, y la capa petición/respuesta que se encarga de las interacciones utilizando distintos métodos. El cambio en la capa de transporte es muy importante pues el uso de UDP lo hace más eficiente, aunque TCP es más fiable; no obstante, la división de dos subcapas del protocolo CoAP (figura 8) y los distintos tipos de mensajes añaden un mecanismo de transferencia fiable que permite prescindir de TCP.

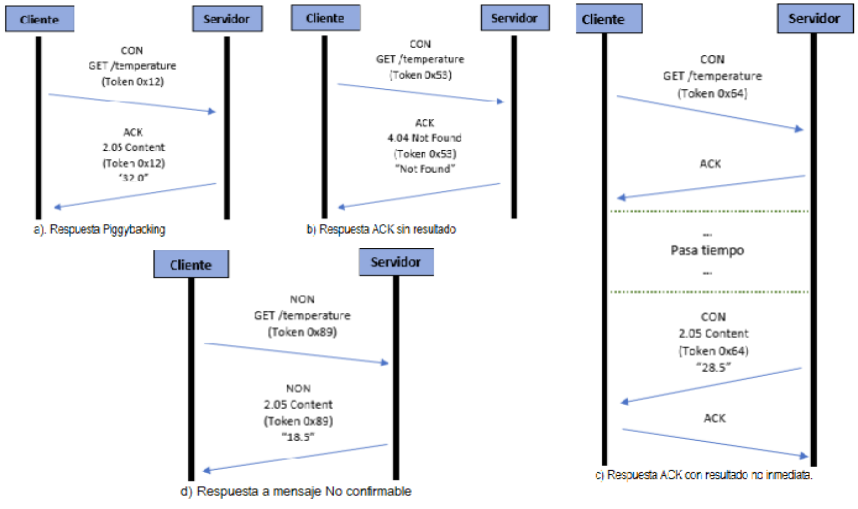
Petición/respuesta

CoAP se implementa en un modelo cliente-servidor, el servidor pone a disposición de los clientes los recursos mediante una URL, que es la dirección única y específica de cada página o recurso que existe en la web y los clientes acceden a ellos utilizando los métodos POST, GET, DELETE y PUT. Las peticiones pueden transmitirse en mensajes confirmables o no confirmables. Un servidor CoAP recibe e interpreta una petición, luego genera una respuesta y se enlaza a un *token* que se generó en el cliente. Las respuestas de un servidor pueden enviarse de diversas formas para mensajes confirmables:

a) Respuesta de *Piggybacking*, el servidor responde una petición con el envío de un ACK junto con la respuesta.
b) Respuesta del ACK sin contenido o sin resultado de la petición.

c) Respuesta del servidor que NO dispone de información. El servidor no puede responder inmediatamente la petición y responde con un mensaje ACK para que el cliente no retransmita la petición; una vez que tenga la respuesta, el servidor la envía al cliente en otro mensaje confirmable, a lo que el cliente responde con un ACK.
d) Si la solicitud es un mensaje no confirmable, la respuesta la envía el servidor sin un ACK.

Figura 9. Tipos de respuestas a mensajes confirmables y no confirmables CoAp



Fuente: Elaboración propia, adaptado de Kowalewski (2018).

Otra respuesta, además de las ya mencionadas, sucede cuando la petición no pueda procesarse por el receptor, en cuyo caso la respuesta es un reinicio (un mensaje RST); este mensaje indica recepción de la petición, pero sin todos los elementos necesarios para generar u obtener una respuesta. Las transacciones petición/respuesta son transparentes para el usuario, y es el protocolo CoAP quien se encarga de manejar el asincronismo.

Mensajes CoAP

A diferencia de HTTP, CoAP utiliza cabeceras de longitud fija de cuatro bytes, se ocupa de los intercambios de información de forma asincrónica a través de UDP, característica que se considera una mejora respecto a HTTP que se logra mediante distintos tipos de mensajes identificables cada uno con un ID, esto permite detectar duplicación de mensajes y tienen una calidad de servicio asociada y de fiabilidad ligera, no como la que proporciona TCP.

Los tipos de mensajes de CoAP son cuatro: confirmable (CON), no confirmable (NON), asentimiento (ACK) y reinicio (RST).

El mensaje confirmable permite transmisiones fiables, utilizando al momento de transmitir un tiempo de espera definido y un tiempo de reenvío exponencial entre las retransmisiones sucesivas hasta que el receptor envía un ACK. Un mensaje no confirmable se utiliza cuando no es necesaria una transmisión fiable, estos mensajes no reciben respuesta ACK, pero sí se pueden detectar mensajes duplicados y también responde con un mensaje Reinicio si no son capaces de procesar el mensaje origen.

Respecto a la seguridad, CoAP está protegido mediante Datagram TLS (DTLS) sobre UDP [RFC6347]. Para realizar transmisiones CoAP seguras entre el cliente y un servidor, se da un proceso de intercambio de información privada para la apertura de la sesión y el intercambio de los datos.

Los sistemas CoAP son interoperables con sistemas HTTP utilizando Proxys entre ambos protocolos. Proxy CoAP-HTTP: permite a los clientes CoAP acceder a los recursos en servidores HTTP a través de un intermediario. Proxy HTTP-CoAP: permite a los clientes HTTP acceder a recursos en servidores CoAP a través de un intermediario.

Servicio de distribución de datos (DDS)

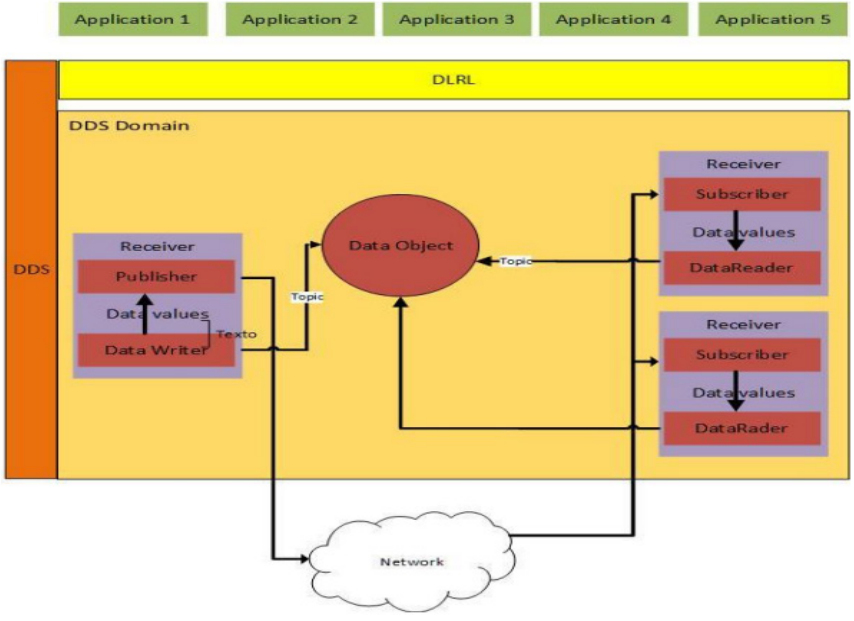
El estándar DDS está definido por el OMG (*Object Management Group*) como una de las mejores opciones para los sistemas distribuidos en tiempo real y proporciona una solución al paradigma centrado en datos en estos sistemas distribuidos con aplicaciones enfocadas a IoT e I-IoT (*Industrial Internet of things*). DDS fue creado en

el 2011 para solucionar los problemas de la industria que requería de una forma de estandarizar los sistemas centrados en datos, que a su vez tuviera un cierto nivel de abstracción que requieren los sistemas distribuidos en tiempo real (González y López, 2017).

A diferencia de otras tecnologías cuya función se centra en el envío de mensajes (*message-centric*), el estándar DDS es un desarrollo enfocado a los datos (*data-centric*) y, aunque ambas parecen ser lo mismo al proporcionar conectividad en sistemas distribuidos, se diferencian en la forma en que lo hacen. Para comprenderlo desde otra perspectiva, el principal objetivo de los sistemas centrados en los mensajes es la entrega de estos sin tener en cuenta su contenido; por otro lado, en un sistema centrado en los datos son prioridad, precisamente, los datos de los usuarios. Bajo este concepto, DDS tiene como ventaja que los datos compartidos proporcionan un alto nivel de abstracción para el usuario; por lo tanto, su unidad de intercambio para este estándar es el valor del dato. De esta manera el sistema comprende el contexto del dato y se encarga de garantizar que todos los suscriptores tengan una visión correcta y consistente del intercambio de información, algo muy similar a una base de datos al proporcionar un esquema global de los datos para gestionar su acceso (Pajuelo et al., 2021).

DDS utiliza un espacio global de datos en el cual los productores escriben en el espacio de datos para que los suscriptores puedan leerlos, esto marca una diferencia con MQTT o AMQP ya que DDS no requiere de un *broker* para su modelo de comunicación, como se aprecia en la figura 10. Trabaja mediante *multicast* con soporte de 23 políticas de QoS que le permite ofrecer a las aplicaciones DDS diferentes criterios de comunicación, seguridad, prioridad, durabilidad y fiabilidad para los desarrolladores de soluciones IoT o I-IoT basadas en M2M (González y López, 2017).

Figura 10. Arquitectura DDS



Fuente: González y López (2017).

Como se aprecia en la figura 10, el modelo de la arquitectura DDS define dos niveles: el DCPS (*Data-Centric Publish-Subscribe*) y el DLRL (*Dats-Local Reconstruction Layer*). El primero de ellos se encarga de entregar la infiltración a los *subscribers*, mientras que el DLRL es opcional y hace las funciones de interfaz para el DCPS, facilitando el intercambio de datos distribuidos entre los objetos (Pajuelo et al.,2021). Para el flujo de datos en el DCPS se manejan cinco entidades:

1. Un *Publisher* que organiza los datos.
2. *DataWriter* que se utiliza por las aplicaciones para interactuar con el *Publisher*.
3. *Subscriber* se encarga de recibir los datos publicados para entregarlos a la aplicación solicitante.
4. *DataReader,* utilizado por el *Subscriber* para tener acceso a los datos recibidos.
5. Tópico, el cual se define como un identificador por tipo de datos y nombre.

A la fecha, DDS continúa en desarrollo y evolución, como es el caso de los denominados X-*Topics* que pueden generarse en tiempo de ejecución y no solamente son definidos por el lenguaje de identificación (IDL). Otros trabajos están enfocados en la integración de tecnologías web como HTML, clientes SOAP (Protocolo de Acceso a Objeto Simple) o REST (Transferencia de Estado Representacional) (Pajuelo et al., 2021). Algunas de sus aplicaciones representativas son en el control de tráfico aéreo, gestión de transportes, salud, energía y *Big Data*, todas ellas dentro de los ámbitos del IoT e I-IoT (González y López, 2017).

Message Queue Telemetry Transport (MQTT)

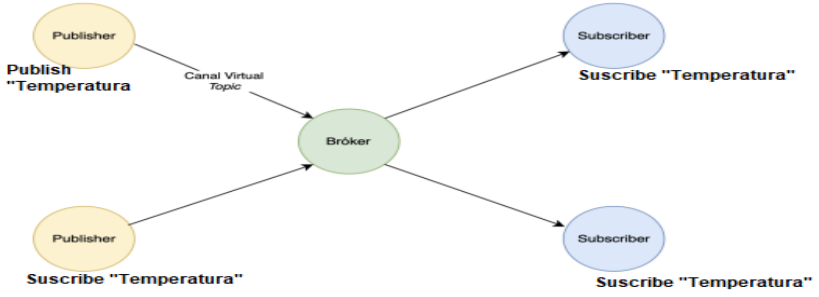
MQTT es un protocolo abierto, ligero y eficiente, propio para dispositivos de pocos recursos como microcontroladores pequeños, facilita la comunicación entre el dispositivo y la nube, y entre la nube y el dispositivo o muchos dispositivos. Pueden conectarse millones de dispositivos a través de este protocolo M2M, apto para aplicaciones IoT. MQTT se convirtió en el estándar OASIS, está pensado para redes con poco ancho de banda, alta latencia y poco fiables, es un protocolo orientado a mensajes, los cuales tienen un encabezado fijo de 2 o 5 bytes, un encabezado opcional y hasta 256 Mbs trabaja con TCP en la capa de transporte y permite implementar seguridad, pues el *broker* puede solicitar autenticación al cliente y admite tres niveles de Calidad de Servicio (QoS) y garantizando así la entrega del mensaje:

- QoS 0 *unacknowledged* (*at most one*): El mensaje se envía una única vez. En caso de fallo puede que algún mensaje no se entregue.
- QoS 1 *acknowledged* (*at least one*): El mensaje se envía hasta que se garantiza la entrega. En caso de fallo, el suscriptor puede recibir mensajes duplicados.
- QoS 2 assured (*exactly one*). Se garantiza que cada mensaje se entrega al suscriptor, únicamente una vez.

MQTT es un protocolo que trabaja basado en eventos y en el modelo publicador/suscriptor, la arquitectura de este protocolo la conforman dos tipos de sistemas: cliente y *broker* (ver Figura 11).

Un *broker* es el servidor que recibe mensajes del cliente, los filtra por tema y los comunica con los demás clientes; es decir, recibe comunicación de unos y se las envía a otros. Los clientes no se comunican entre sí, pues reciben la comunicación del *broker*. Un cliente puede ser editor, suscriptor o ambos. La comunicación entre los dos sistemas solo se lleva a cabo cuando hay información o nuevos datos para enviar (Jaffey, 2014).

Figura 11. Arquitectura típica MQTT



Fuente: Elaboración propia, adaptada de Jaffey (2014).

Los mensajes se organizan jerárquicamente como temas y se filtran para enviarse a cada cliente. Un cliente puede publicar temas, otros clientes pueden suscribirse a estos temas y el *broker* le hará llegar los mensajes correspondientes a cada suscriptor.

En MQTT se efectúan cuatro acciones básicas de comunicación:

- Publicar: son los datos específicos de la implementación que se envía a los suscriptores. Si un tema publicado no existe el *broker* lo crea.
- Suscribirse: un cliente envía un paquete SUBSCRIBE para convertirse en suscriptor de un tema o mediante comodines se puede suscribir a ramas completas de temas, como respuesta recibe un paquete SUBACK y, si hay mensajes retenidos del tema, también recibe dichos mensajes.
- Ping: el envío de paquetes PINGREQ del cliente al *broker*, recibe un paquete de PINGRESP como verificación de la conectividad, un *ping* se puede utilizar para garantizar que la sesión TCP siga funcionando.

- Desconectar: se puede enviar un paquete DISCONNECT al *broker*, con esto el *broker* interpreta que ya no es necesario enviar o poner en cola los mensajes para un suscriptor y que ya no recibirá.

Conclusiones

Una vez expuestos los protocolos IoT en cada una de las capas del modelo propuesto por la IETF, se observa que es posible lograr la comunicación sin la necesidad de que todos tengan implementado el modelo TCP/IP. Sin embargo, la transferencia y visualización de la información recopilada por los dispositivos IoT requiere que solo unos cuantos dispositivos hagan la funcionalidad de gateway hacia el mundo IP. La gran diversidad de nodos IoT complica su implementación debido a que no se han logrado establecer arquitecturas físicas y lenguajes de programación estandarizados que faciliten la interoperabilidad de todos los protocolos analizados. No obstante, el desarrollo e implementación de esta tecnología se ha disparado de manera exponencial en la actualidad, prueba de ello son los diversos protocolos expuestos para cada una de las capas IoT analizadas. Asimismo, como sucede en las redes tradicionales IP, se debe tener especial cuidado en diferenciar algunos protocolos de comunicaciones que tienen el nombre de la aplicación o tecnología que lo utiliza.

Referencias

Abhimanyu, P. (2019). What is Power Line Communication (PLC) and how it works. *CircuitsDigest*. https://circuitdigest.com/article/what-is-power-line-communication-plc-and-how-does-it-work

Bassi, A. (2021). Introducción al Protocolo CoAP. *Goto IoT*. https://www.gotoiot.com/pages/articles/coap_intro/index.html

Caiza, G., Edison, M., Remache, E., Ortiz, A. y García, M. V. (2020). Comparación de AMQP y CoAP para la integración de las comunicaciones en el área de producción. *Revista Iberica de Sistemas e Tecnologias de Informacao, 2020(E26),* pp. 652-666. https://pure.ups.edu.ec/es/publications/comparaci%C3%B3n-de-amqp-y-coap-para-la-integraci%C3%B3n-de-las-comunicacio

Egli, P. (2015). LPWAN. Overview of emerging technologies for Low Power Wide Area Networks in Internet of things and M2M scenarios. *Indigoo*. http://indigoo.com/dox/itdp/12_MobileWireless/LPWAN.pdf

García, L. (2020), ¿Cuáles son las arquitecturas y componentes de una red IoT? *Redes Móviles*. https://redesmoviles.com/iot/arquitecturas-iot/

González, A. y López, J. (2017). IoT: dispositivos, tecnologías de transporte y aplicaciones [Tesis de maestría, Universitat Oberta Catalunya]. Repositorio Institucional UOC.

Harwood, T. (2021). IoT standards and protocols. *Postscapes Tech*. https://www.postscapes.com/internet-of-things-protocols/

Hrasnica, H., Haidine, A., Lehnert, R. (2001). Performance comparison of reservation MAC protocols for broadband powerline communications networks. *Revista Internet Performance and Control of Network Systems II*, pp. 302-313.

Hussein T. M., Melike, E. (2015). Comunicaciones de redes inteligentes: oportunidades y Desafíos. En M. S. Obaidat, A. Anpalagan e I. Woungang (Eds.), *Handbook of green information and communication systems*. https://doi.org/10.1016/B978-0-12-415844-3.00025-5

Internet Engineering Task Force (IETF). (2021a). CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets. *IETF Drafts*. https://tools.ietf.org/id/draft-ietf-core-coap-tcp-tls-11.html

Internet Engineering Task Force (IETF). (2021b). Group Communication for the Constrained Application Protocol (CoAP), Internet Engineering Task Force RFC 7390-2014. 56. *IETF Drafts*. https://datatracker.ietf.org/doc/html/rfc7390

Internet Engineering Task Force (IETF). (2021c). Block-Wise Transfers in the Constrained Application Protocol (CoAP), Internet Engineering Task Force RFC 7959-2016. *IETF Drafts*. https://datatracker.ietf.org/doc/html/rfc7959

Internet Engineering Task Force (IETF). (2021d). CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets, Internet Engineering Task Force RFC 8323-2018. *IETF Drafts*. https://www.rfc-editor.org/rfc/pdfrfc/rfc8323.txt.pdf

IONOS. (2019). *AMQP: Conoce el Advanced Message Queuing*. Digital Guide IONOS. https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/advanced-message-queuing-protocol-amqp/

ISO/IEC. (2014). *Information technology Advanced Message Queuing Protocol (AMQP) v1.0 specification.* Suiza: ISO/IEC 19464:2014. https://www.iso.org/standard/64955.html

Jaffey, T. (2014). *MQTT and CoAP, IoT Protocols.* Eclipse Foundation. https://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php

Kowalewski, O. (2018). IoT: Evaluación de protocolos limitados de nivel de aplicación para Internet de las cosas [Tesis de Maestría, Universidad Carlos III de Madrid]. Repositorio Institucional UC3M.

Lora. (2015). LoRaWan What is it? *Lora-Alliance.* https://lora-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf

López, G., Matanza, J., De la Vega, D., Castro, M., Arrinda, A Moreno, J. y Sendin, A. (2019). The role of PowerLine Communications in the smart grid revisited: applications, challenges, and research initiative. *Revista IEEE Access*, 99, pp. 117346-117368. http://dx.doi.org/10.1109/ACCESS.2019.2928391

Microsoft. (2021). *Protocolos y tecnologías de IoT*. Azure. https://azure.microsoft.com/es-mx/overview/internet-of-things-iot/iot-technology-protocols/

Moes, T. (2020). *¿Qué es LTE? Diferencias entre LTE y 4G*. Software Labs. https://softwarelab.org/es/lte-4g/

OASIS. (2021). OASIS Advanced Message Queuing Protocol (AMQP) Version 1.0. *OASIS Open*. http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-complete-v1.0-os.pdf

Pajuelo, R., González, R., Hidalgo, E. (2021). Estudio de la arquitectura DDS y diseño de una solución aplicada sobre arquitectura genérica de vehículos. (Trabajo Fin de grado inédito). Universidad de Sevilla.

Pardal-Garcés, P. (2017) Redes de área extensa para aplicaciones IoT: modelado de comunicaciones Sigfox (Tesis de máster, Universidad Politécnica de Valencia). https://riunet.upv.es/handle/10251/86052

D'Alessandro, S. y Tonello, A. (2014). Relaying protocols for In-Home PLC. En En L. Torsten, A. Schwager, P. Pagani y D. Schneider (Eds.), *MIMO Power Line Communications* (pp. 553-572). CRC Press. https://doi.org/10.1201/b16540

Tameem, H. (2021). Overview of the future Internet architecture projects and protocols and applications. [Tesis de grado, Nanjing University of Science and Technology]http://dx.doi.org/10.6084/m9.figshare.14938434.v1

ZigBee. (2021). *IEEE 802.15 Working group for Wireless Specialty Networks (WSN). IEEE: 802.15.* https://www.ieee802.org/15/

ZigBee-2 (2021). *ZigBee Specification 05347r17.* CSA Alliance. www.zigbee.org.

Capítulo 6

Plataformas para IoT

José Román Herrera Morales
Armando Román Gallardo
Sara Sandoval Carrillo
Margarita Glenda Mayoral Baldivia

Resumen

En este capítulo se aborda la definición de lo que entendemos por plataformas IoT, cuáles son las principales características que tienen estas plataformas y cómo se clasifican en diferentes tipos de según la orientación de los servicios que proporcionan. También, se realiza una comparativa de cuáles son las mejores plataformas de acuerdo con la selección que llevan a cabo portales especializados como *Dzone.com*, *IoTforAll.com*, *G2.com* y *sam-solutions.com*. Asimismo, se incluye una clasificación adicional denominada *plataformas industriales para* IoT (I-IoT) y un recuento de las mejores plataformas I-IoT según la empresa Gartner. Finalmente, se incluye un *ranking* donde se muestran una selección de las 20 plataformas IoT que destacaron como las mejores, considerando las diferentes comparativas presentadas.

Introducción

El gran crecimiento que está teniendo el IoT en los años recientes ha propiciado que se estén generado un sinfín de nuevos aparatos, *gadgets*, sensores, dispositivos inteligentes, así como plataformas y servicios en la nube para ofrecer aplicaciones especializadas tanto para la industria, el comercio, el gobierno y hasta para el usuario final. En un reporte reciente del portal Markets and Markets (M&M, 2020) analizan este comportamiento y estiman una tasa de crecimiento anual del 12.6% durante el periodo de 2020 a 2025; asimismo, se prevé un impacto de forma similar en el crecimiento, tanto en la oferta de plataformas IoT como en empresas proveedoras de estos servicios, esto basado en que en 2019 se conocían públicamente 620 plataformas IoT, cifra que representa casi el doble a las existentes en el 2015 (Statista, 2019). Este crecimiento observado, nos da una idea de la gran diversidad de plataformas IoT que se tienen disponibles en la actualidad y el surgimiento esperado de nuevas plataformas que se tendrán en los próximos años, por lo que resulta muy importante adentrarnos en estos temas, a fin de conocer más sobre las plataformas para IoT.

¿Qué es una plataforma IoT?

Si se trata de establecer una definición de lo que es una plataforma IoT, pudiéramos abordarlo desde diferentes perspectivas, que van desde el tratamiento que le dan en la literatura científica, donde se suele analizar a un nivel más detallado de capas, componentes y sus interconexiones, hasta pasar por un enfoque más integral como el que se le da en la industria y mercado de fabricantes (Asemani, 2019), en el cual se maneja como una oferta de servicios de gestión central que pueden incluir herramientas y plataformas para desarrollo y procesamiento.

Así, una plataforma IoT puede ser vista como la integración de componentes de *hardware*, *software*, una capa de interconectividad, un espacio de almacenamiento de datos y las aplicaciones diseñadas específicamente para que el usuario final pueda interactuar, leer o interpretar los resultados del procesamiento de la información que fue recolectada por los sensores de IoT (Sakovich, 2021).

También, se puede definir una plataforma IoT como una tecnología multicapa que facilita la comunicación máquina a máquina (M2M) y que se utiliza para administrar, automatizar y poner en línea una gran diversidad de dispositivos (SoftwareTestingHelp, 2021). Ante la gran diversidad de fabricantes y plataformas existentes, lo importante es identificar el tipo de plataforma de la que se trata, así como cuáles son los principales servicios y prestaciones que pueden ofrecerle al usuario.

Tipos de plataformas IoT

Las plataformas IoT se pueden clasificar en diversos tipos, dependiendo de la tarea en la que se enfocan los servicios que proporcionan, de esta manera se tienen los siguientes tipos de plataformas IoT (Sakovich, 2021):

- Plataformas para *desarrollo de hardware*
- Plataformas para el *desarrollo de apps*
- Plataformas de *conectividad*
- Plataformas de *analítica de datos*
- Plataformas *end-to-end*

Las plataformas IoT para desarrollo de *hardware* proporcionan interfaces y emuladores para diseñar e interconectar los circuitos electrónicos con placas de desarrollo con una amplia gama de sensores y actuadores, son muy útiles para probar soluciones de IoT a nivel de *hardware*. También suelen combinarse e integrarse con plataformas de desarrollo de aplicaciones que proporcionan un entorno de desarrollo integrado (IDE) que facilitan en gran medida el programar o codificar las aplicaciones.

Las plataformas de conectividad proporcionan el soporte de la capa de red y telecomunicaciones para transmitir información entre los dispositivos físicos y la capa de procesamiento que bien puede estar ubicada de forma local y centralizada o en la nube, para esto se pueden utilizar protocolos y estándares de conectividad para IoT que van desde HTTP hasta protocolos especializados como MQTT (MQ *Telemetry Transport*), DDS (*Data Distribution Service*), AMQP (*Advanced Message Queing Protocol*), CoAP (*Constrai-*

ned Application Protocol), Bluetooth, BLE (Bluetooth *Low Energy*), ZigBee, Wi-Fi, SigFox, Cellular, LoRA, LoRaWAN, entre otros (Silva, 2019). Para más detalles de protocolos de interconectividad para IoT se puede revisar el capítulo 5.

Las plataformas *end-to-end* se basan en servicios en la nube y son soluciones integrales puesto que ofrecen los servicios y características apropiados para cubrir las necesidades de desarrollo de aplicaciones para IoT de extremo a extremo, es decir, cubren todas las etapas y van desde los procesos de adquisición de datos, su almacenamiento y procesamiento hasta las etapas de analítica y visualización de datos, todo ello a través de una interfaz integrada y de fácil manejo.

Plataformas Industriales para IoT (I-IoT)

Además de los tipos de plataformas IoT que se mencionaron en la sección previa, podemos incluir otra clasificación que también es muy utilizada para describir plataformas especializadas para aplicaciones del IoT, estas son las *plataformas industriales del Internet de las cosas (*I-IoT*)*. De acuerdo con Gartner (2021), se define como una *plataforma I-IoT* al conjunto de capacidades de *software* integradas para abarcar los esfuerzos para mejorar la toma de decisiones de gestión de activos, así como la visibilidad y el control operativo de plantas, depósitos, infraestructura y equipos dentro de industrias intensivas en activos. En sentido estricto, esta clasificación de plataformas I-IoT puede considerarse similar a las plataformas *end-to-end*, por lo que no debería extrañarnos que diversas plataformas comerciales puedan aparecer en una u otra clasificación, si revisamos diversas fuentes de consulta.

Las plataformas I-IoT pueden utilizarse como una suite de tecnología o como una plataforma de aplicaciones abiertas y de uso general, o incluso utilizar diversas combinaciones de implementación que mejor se amolden a las necesidades del usuario. Estas plataformas son desarrolladas por empresas líderes de la industria del *software* (Google, Amazon, Microsoft, Oracle, IBM, SAP, etc.) y también por empresas del ámbito empresarial (como PTC, GE, Samsung, Siemens, Hitachi) que tienen gran impacto y tra-

dición puesto que han venido ofreciéndoles a sus clientes desde hace varios años, soluciones de gestión que han ido evolucionando hasta llegar a esta gama de servicios que se integran a la infraestructura existente.

Estas plataformas I-IoT extienden sus capacidades con dispositivos IoT como sensores, controladores, enrutadores, puntos de acceso, puertas de enlace y sistemas que se ubican en infraestructuras en la nube (*cloud*) o en variantes como el cómputo en el borde (*edge computing*) o cómputo en la niebla (*fog computing*) para aprovechar los avances tecnológicos en las áreas de inteligencia artificial (*artificial intelligence*) y analítica de negocios (*business intelligence*). Estas plataformas cumplen con estándares muy estrictos de la industria y están diseñadas para cumplir los requisitos de seguridad, de protección y aplicaciones de misión crítica activos industriales y sus entornos operativos.

Top I-IoT de acuerdo con Gartner's *Magic Quadrant* 2021

En este contexto de plataformas industriales para IoT, la empresa consultora Gartner ha venido publicando en los últimos años un reporte anual denominado *Magic Quadrant para plataformas industriales de IoT* (Gartner-MQ, 2021), donde se analiza y clasifica a las diversas plataformas considerando el cumplimiento de varios indicadores dentro de dos dimensiones que definen como son su habilidad para ejecutar (*ability to execute*) e integridad de su visión empresarial (*completeness of vision*). Con los resultados de este análisis se genera una gráfica (mostrada en la Figura 1) conformada por cuatro bloques o cuadrantes en donde posicionan a las plataformas como líderes (*leaders*), visionarios (*visionaries*), retadores (*challengers*) o jugadores de nichos de mercado (*niche players*).

En la edición 2021 del *Magic Quadrant*, para abreviar MQ2021, se identifican a empresas muy reconocidas del sector de TI como Microsoft y su producto Azure IoT (Microsoft, 2021) así como Amazon y su plataforma AWS IoT Core. También se ubica en una posición destacada a la plataforma ThingWorx de PTC (PTC, 2021), por ser parte del selecto grupo que se encuentra en el cua-

drante de líderes. En la figura 1 se aprecia la gráfica original del MQ2021 de Gartner, mientras que en la tabla 1 se incluye la lista completa de las empresas posicionadas en estos cuadrantes.

Figura 1. *Magic Quadrant* 2021
para plataformas industriales de I-IoT

Fuente: Tomado de Gartner-MQ (2021).

Tabla 1. Empresas y plataformas incluidas en el *Magic Quadrant* 2021 para plataformas industriales de IoT (I-IoT)

Cuadrante	Empresa y plataforma	Website
Líderes (*leaders*)	Microsoft Azure IoT	https://azure.microsoft.com/en-us/overview/iot/
	PTC ThingWorx	https://www.ptc.com/en/products/thingworx
	Hitachi Vantara	https://www.hitachivantara.com/es-latam/home.html
	Software AG Cumulocity	https://www.softwareag.cloud/site/product/cumulocity-iot.html/
Retadores (*challengers*)	Amazon AWS IoT	https://aws.amazon.com/es/iot/
Jugadores de nichos de mercado (*niche players*)	Samsung SDS	https://www.samsungsds.com/en/iot/iot.html
	Davra IoT Platform	https://davra.com/
	Litmus Edge	https://litmus.io/
	ABB Ability Genix	https://new.abb.com/cpm/industrial-software-solutions/genix
	Flutura Cerebra	https://www.flutura.com/cerebra
	Exosite Murano	https://www.exosite.com/murano-iot-platform
	Braincube Smart IIoT Platform	https://braincube.com/solutions/iiot-platform/
	Altizon Datonis	https://altizon.com/datonis-iiot-platform/
	Rootcloud ROOTCLOUD	https://en.rootcloud.com/
	Envision Digital -Envision Operating System (EnOS)	https://www.envision-digital.com/enos-platform/
	Eurotech Everyware	https://www.eurotech.com/en/products/iot
	Knowledge Lens iLens	https://www.knowledgelens.com/Products/iLens/
Visionarios (*visionaries*)	Siemens MindSphere	https://siemens.mindsphere.io/en

Fuente: Elaboración propia.

Las mejores plataformas para IoT

Ante la amplia oferta de plataformas IoT, se ha realizado una revisión en portales especializados buscando cuáles son las plataformas mejor ranqueadas para identificar aquéllas que aparecen consistentemente en estos listados de las Top Plataformas IoT, para ello:

- Se ha considerado el listado de las *Top 10 plataformas de IoT para el 2021* (Shten, 2020) del portal *Dzone.com*, el cual tiene una sección especializada llamada *IoT Zone*, donde publican artículos de análisis y de opinión, así como tutoriales sobre uso de plataformas y herramientas para IoT.
- En el portal *G2.com,* que se especializa en el análisis de diversos productos y tecnologías de *software*, se ofrece una clasificación de plataformas IoT de acuerdo con un método de *rankeo* propio donde se considera el nivel de satisfacción de los usuarios basado en información recopilada del web, de redes sociales y del análisis de comentarios realizados por los usuarios de cada plataforma en los que ellos evalúan entre otras cosas: la facilidad de uso, de configuración y de administración, la calidad de soporte técnico, y también la facilidad con la que pueden hacer negocios con ella. En este portal se ofrece una lista del *Top 6 plataformas IoT mejor rankeadas* (G2.com, 2021)
- Se han considerado los listados del top de plataformas IoT en los portales web especializados: *IoTforAll.com (Janson, 2021)*, *SaM-solutions.com (Sakovich, 2021)* y *SoftwareTestingHelp.com (2021)* a fin de completar esta comparativa de las plataformas más relevantes para IoT.
- También se han tomado en cuenta las plataformas I-IoT que están incluidas en el MQ2021 de Gartner.

Dzone.com – Top 10 plataformas IoT

1. Google Cloud IoT
2. AWS IoT
3. PTC ThingWorx
4. Kaa IoT Platform
5. Deutsche Telekom
6. Salesforce
7. Cisco IoT networking and Edge Intelligence
8. Microsoft Azure IoT
9. Radix IoT
10. GE Predix

G2.com – Top 6 plataformas IoT

1. Particle
2. Hologram
3. IBM Watson IoT
4. Google Cloud IoT
5. AWS IoT core
6. PTC ThingWorx

IoTforAll.com – Top 10 plataformas IoT

1. Google Cloud IoT
2. Cisco IoT Cloud Connect
3. Salesforce IoT Cloud
4. IRI Voracity
5. Particle
6. IBM Watson IoT
7. PTC ThingWorx
8. Amazon AWS IoT Core
9. Microsoft Azure IoT Hub
10. Oracle IoT

SoftwareTestingHelp.com – Top 12 plataformas IoT

1. Google Cloud Platform
2. OpenRemote
3. IRI Voracity
4. Particle
5. PTC ThingWorx
6. IBM Watson IoT
7. Amazon AWS IoT Core
8. Microsoft Azure IoT Suite
9. Oracle IoT
10. Cisco IoT Cloud Connect
11. Altair SmartWorks
12. Salesforce IoT Cloud

Otras plataformas de IoT para desarrollo y experimentación

Por último, se presenta una selección de otras plataformas de IoT que son útiles para experimentación, autoaprendizaje, proyectos en el contexto académico, ya sea para para facilitar en desarrollo de apps, para proyectos escolares, o para tener un primer contacto en proyectos de *hobbie*. Varias de estas plataformas tienen versiones gratuitas de prueba y algunas son del tipo *open-source*, por lo que pueden utilizarse sin ninguna restricción. A continuación, un listado de estas plataformas para IoT:

- Zetta JS - https://www.zettajs.org/
- Node-RED - https://nodered.org/
- ThingsBoard - https://thingsboard.io/
- Kaa IoT Platform - https://www.kaaiot.com/
- Thinger - https://thinger.io/
- Ubidots - https://ubidots.com/
- Blink IoT platform - https://blynk.io/

Rankeo de las mejores plataformas IoT

Considerando las listas de mejores plataformas de los portales mencionados en la sección anterior, más la selección de las plataformas que aparecen en los cuadrantes del MQ2021 para plataformas industriales de IoT, se ha elaborado un método de *rankeo* propio para identificar a las mejores 20 plataformas de IoT. En la tabla 2 se muestran de forma ordenada las plataformas IoT que ocuparon los primeros lugares de acuerdo con este *ranking*.

Tabla 2 . Top 20 de plataformas para IoT con la comparativa de posiciones

Plataforma IoT	SoftwareTestingHelp.com	Dzone.com	G2.com	IotForAll.com	Gartner MQ2021	*Ranking*
Google Cloud IoT	1	1	4	1	-	1
PTC ThingWorx	5	3	6	7	Leaders	2
Amazon AWS IoT Core	7	2	5	8	Challengers	3
Particle	4	-	1	5	-	4
Microsoft Azure IoT	8	8	-	9	Leaders	5
IBM Watson IoT	6	-	3	6	-	6
Cisco IoT Cloud Connect	10	7	-	2	-	7
IRI Voracity	3	-	-	4	-	8
Hitachi Vantara	-	-	-	-	Leaders	9
Software AG Cumulocity	-	-	-	-	Leaders	10
OpenRemote	2	-	-	-	-	11
Hologram	-	-	2	-	-	12
Siemens MindSphere	-	-	-	-	Visionaries	13
Kaa IoT Platform	-	4	-	-	-	14
Salesforce IoT Cloud	12	6	-	-	-	15
Deutsche Telekom	-	5	-	-	-	16
Oracle IoT	9	-	-	10	-	17
Radix IoT	-	9	-	-	-	18
GE Predix	-	10	-	-	-	19
Altair SmartWorks	11	-	-	-	-	20

Fuente: Elaboración propia.

Para aplicar este *rank* se consideraron varios criterios, entre ellos: 1) que la respectiva plataforma IoT aparezca en más de un listado, 2) la posición que obtuvo en las listas Top individuales de cada portal, es decir, no es lo mismo que una plataforma sí aparezca en varias listas, pero sea la décima posición, a que aparezca solamente en dos listados pero que en éstos ocupe uno de los primeros cinco lugares. También, se agregó un puntaje adicional a aquellas plataformas que aparecieron en los cuadrantes del MQ2021 de Gartner. Cabe mencionar que en la tabla 2 no aparecen todas las plataformas de los diferentes listados previos, puesto que se ha limitado a seleccionar el Top 20 de plataformas IoT. Otro detalle importante para destacar es que, dentro del *ranking*, tres de las primeras cinco plataformas están presentes en los cuadrantes del MQ2021 de Gartner.

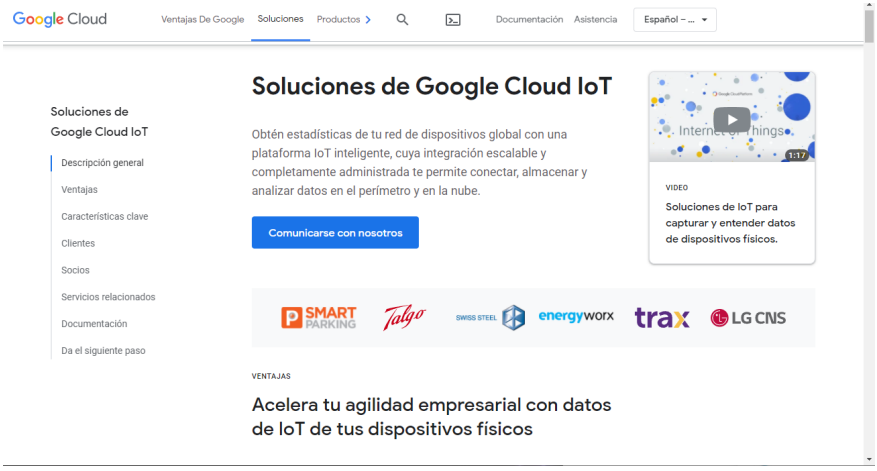
TOP 3 plataformas IoT

A continuación, se presenta una breve descripción de las plataformas que resultaron en los tres primeros lugares del *ranking* de plataformas IoT que presentamos en la tabla 2:

Google Cloud IoT

Es la plataforma especializada en IoT de la empresa Google, del tipo *end-to-end*. Tiene el respaldo de uno de los grandes actores mundiales de la industria de TI y cuenta con una gran integración con su tecnología en la nube Google Cloud Platform (GCP). Se definen como una plataforma IoT inteligente, con integración escalable y autoadministrada que ofrece servicios para conectar, almacenar y analizar la información tanto en el borde como en la nube (*fog and cloud computing*). Es destacable que, aunque esta plataforma de Google no aparece en el listado de MQ2021 de Gartner, en la revisión que se realizó en los otros portales especializados siempre estuvo en los primeros lugares, por lo que se ha ganado su lugar en esta selección que presentamos.
El portal de Google Cloud IoT es: https://cloud.google.com/solutions/iot (ver figura 2).

Figura 2. Portal web de la plataforma Google Cloud IoT



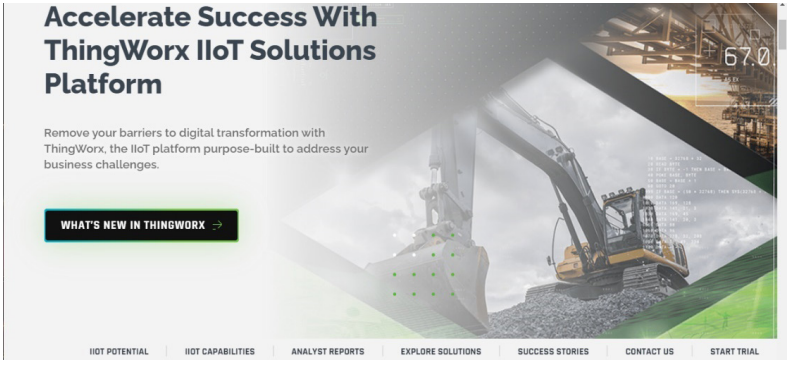
Fuente: Elaboración propia, captura de pantalla.

PTC ThingWorx

PTC (anteriormente *Parametric Technology Corporation*) es una empresa estadounidense fundada en 1985, que quizás en el mundo de las TI no sea tan conocida como Google, Amazon o Microsoft. Sin embargo, en el contexto industrial es ampliamente reconocida por sus aplicaciones de gestión de producción, diseño industrial (CAD), realidad aumentada (AR) y su plataforma industrial para el IoT ThingWorx.

Para explorar esta plataforma y obtener información más detallada puede visitar su sitio web: https://www.ptc.com/en/products/thingworx, en la figura 3 se presenta la pantalla de su portal web.

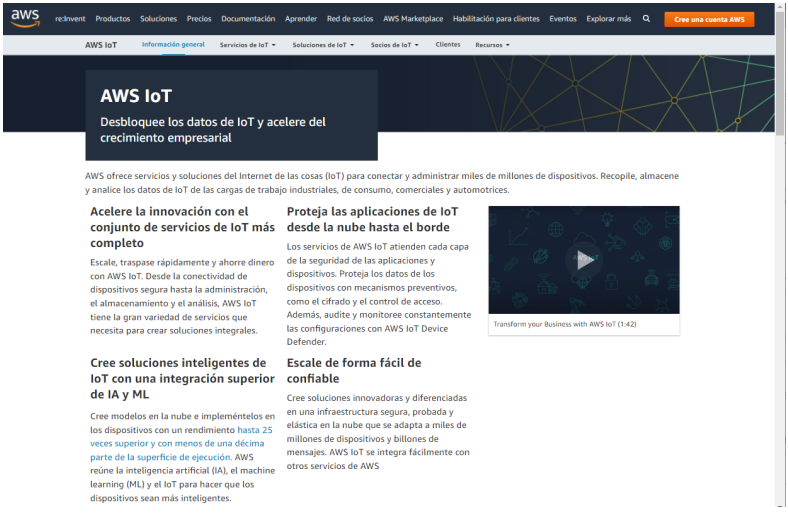
Figura 3. Portal web de la plataforma ThingWorx de PTC



Amazon AWS IoT Core

Amazon es sin duda un líder indiscutible del mercado en servicios basados en la nube, su plataforma IoT Core ofrece a todo tipo de usuario gestionar aplicaciones para IoT, integrando de forma escalable capas de inteligencia artificial (IA) y modelos de *machine learning* (ML) para obtener soluciones inteligentes. La figura 4 muestra la pantalla de inicio de su portal web.

Figura 4. Portal web de la plataforma de Amazon AWS IoT Core



Fuente: Elaboración propia, captura de pantalla.

Para conocer más detalles y explorar esta plataforma puede acceder a su sitio web: https://aws.amazon.com/es/iot/

Conclusiones

En este capítulo hemos revisado el tema de las plataformas IoT, proporcionando definiciones de estas plataformas de servicios, así como clasificaciones realizadas con respecto a las principales tareas y servicios en los que se especializan. De estos tipos de plataformas IoT, se destacan las plataformas integrales o *end-to-end,* a las cuales, por la orientación de sus procesos, se les puede conocer como *plataformas industriales para IoT* debido a que cubren las necesidades de este tipo de usuarios, con una gama de servicios que van de extremo a extremo y a través de un entorno integral basado en tecnologías en la nube. También, se ha incluido un *ranking* de las mejores plataformas IoT considerando los listados que publican portales especializados del área como *Dzone.com*, *IoTforAll.com*, *G2.com,* contrastados con los resultados del reporte anual *MagicQuadrant 2021 for Industrial IoT Platforms* que publica la consultora Gartner y, con ello, identificamos el Top 20 de plataformas IoT, destacando a los tres primeros lugares de este *ranking*. Por último, queda mencionar que la industria del IoT sigue en plena expansión y se espera que la oferta de servicios, plataformas, soluciones integrales y una amplia gama de aplicaciones sigan en franco crecimiento en los próximos años.

Referencias

Asemani, M., Abdollahei, F. y Jabbari, F. (2019). Understanding IoT platforms: towards a comprehensive definition and main characteristic description. En *2019 5th International Conference on Web Research (ICWR). IEEE, 2019* (pp. 172-177). https://doi.org/10.1109/ICWR.2019.8765259

Janson, C. (2021). Top 5 IoT development Platforms in 2021. *IoTforAll.com*. https://www.iotforall.com/top-5-iot-development-platforms-in-2021

G2.com. (2020). *The top 6 IoT platforms*. https://www.g2.com/categories/iot-platforms?tab=highest_rated

Gartner. (2021). *Industrial IoT Platforms reviews and ratings*. https://www.gartner.com/reviews/market/industrial-iot-platforms

Gartner. (2021, 18 de octubre). *Magic Quadrant for Industrial IoT Platforms*. https://www.gartner.com/doc/reprints?id=1-27IESWUW&ct=210922&st=sb

Markets and Markets (2020). *IoT cloud platform market by offering (platform and service), deployment mode (public cloud, private cloud and hybrid), organization size, application area (building and home automation and connected healthcare), and region - global forecast to 2025*. *Market Research Report TC 4609*. https://www.marketsandmarkets.com/Market-Reports/iot-cloud-platform-market-195182.html

Microsoft. (2021, 25 de octubre). *Microsoft named a Leader in the 2021 Gartner® Magic Quadrant™ for Industrial IoT Platforms*. https://azure.microsoft.com/en-us/blog/microsoft-named-a-leader-in-the-2021-gartner-magic-quadrant-for-industrial-iot-platforms/

PTC. (2021). *PTC has been recognized as a Leader among IIoT platform providers in 2021 Gartner® Magic Quadrant™ for Industrial IoT Platforms*. https://www.ptc.com/en/resources/iiot/white-paper/gartner-mq-for-iiot

Sakovich, N. (2021). 10 best IoT platforms for 2021. *SaM-Solutions.com*. https://www.sam-solutions.com/blog/top-iot-platforms/

Shten, O. (2020). *10 best IoT platforms for 2021*. *Portal Dzone.* https://dzone.com/articles/10-best-iot-platforms-for-2021

Silva, J.d.C., Rodrigues, J.J.P.C., Al-Muhtadi, J., Rabêlo, R.A.L. y Furtado, V. (2019). Management platforms and protocols for Internet of things: a survey. *Sensors, (19)*3, p. 676. https://doi.org/10.3390/s19030676

SoftwareTestingHelp.com. (2021). 10 best IoT platforms to watch out in 2021. https://www.softwaretestinghelp.com/best-iot-platforms/

Statista (2019). *Number of publicly known Internet of things (IoT) platforms worldwide from 2015 to 2019*. https://www.statista.com/statistics/1101483/global-number-iot-platform/

Capítulo 7

Aplicaciones del aprendizaje automático para el Internet de las cosas

Erika M. Ramos Michel
María Andrade Aréchiga
Jorge Rafael Gutiérrez Pulido
Pedro Damián Reyes
Ricardo Acosta Díaz

Resumen

El Internet de las cosas, también conocido como el Internet de todo, junto con la inteligencia artificial, plataformas *cloud* y tecnologías como *Data Mining*, *Big Data*, y aprendizaje automático nos permiten crear una nueva gama de aplicaciones en un entorno todo-conectado. En este capítulo se presenta una breve descripción, tanto histórica como actual, de los conceptos inteligencia artificial, Internet de las cosas y otros relacionados, así como de algunas aplicaciones de aprendizaje automático.

Introducción

La inteligencia artificial (IA), es una rama amplia de las ciencias de la computación que estudia e investiga formas de imitar el funcionamiento de las neuronas humanas a través de computadoras, con el objetivo de realizar tareas que normalmente requieren inteligencia humana. Estos algoritmos permiten a una máquina interpretar datos y situaciones para responder de una manera determinada, y aprender a la vez, de cada situación. Cuando se habla de IA no debe pensarse en robots, pues hay otras formas de IA con las

que podemos interactuar, por ejemplo, los *chatbots*, con los cuales se puede hablar e incluso jugar (D'Arc, 2020).

Para entender cómo la IA forma parte de nuestro quehacer diario, es necesario tener un panorama general de cómo surgió y evolucionó a lo largo de la historia. Para ello, en esta sección se presenta un breve recuento sobre su evolución.

En el siglo IV a. C., Aristóteles fue uno de los primeros en intentar codificar la manera correcta de pensar, estableció el primer solucionador de problemas estructurado a través de silogismos, esquemas de estructuras de argumentación, que describen respuestas racionales de la mente humana y que, al seguirlos, producen conclusiones racionales. En 1637, René Descartes predice la posibilidad de crear máquinas que piensen por sí mismas. Descartes se considera también el iniciador de la filosofía racionalista moderna. En 1842, la matemática Ada Lovelace programó el primer algoritmo a procesarse por una máquina; el lenguaje de programación Ada se llama así en su honor. En 1847, el matemático George Boole, sentó las bases de la aritmética computacional, razonamiento lógico capaz de sistematizarse ya que definió la lógica proposicional. En 1950, Alan Turing propuso la Prueba de Turing, prueba de comunicación verbal hombre-máquina basada en detectar hasta qué punto la máquina era capaz de emular la inteligencia de un ser humano; en otras palabras, la máquina superaba la prueba si un humano no era capaz de determinar si las respuestas a preguntas planteadas venían o no de una persona. En la actualidad, para que una computadora pase esta prueba no es fácil, pues debe ser capaz de: procesar lenguaje natural, representar el conocimiento, razonar automáticamente y autoaprender (Russell y Norvig, 2008; Ponce Cruz, 2010).

En 1955, John McCarthy, Marvin Minsky, Nathaniel Rochester y Claude Shannon, veteranos de la computación militar, solicitaron a la Fundación Rockefeller financiamiento para un proyecto cuyo objetivo era formalizar y reproducir en una computadora el comportamiento humano inteligente. Así, en 1956, durante la conferencia de Darmouth, se utilizó por primera vez el término de inteligencia artificial, fundándose así una disciplina de investigación cuya conjetura es que todos los aspectos del aprendizaje o

cualquier otra característica de la inteligencia pueden, en principio, describirse con tanta precisión que una máquina puede ser hecha para simularla (Dick, 2019). En 1956 Newell y Simon publican el programa *Logic Theory Machine* considerado el primer programa de inteligencia artificial (Newell y Simon, 1956). Es hasta la década de los 90, que este concepto empieza a adoptarse como lo conocemos hoy en día.

Sin duda alguna, otros aspectos que contribuyeron al desarrollo de la IA fueron: la regla original de aprendizaje del perceptrón, diseño de la primera red neuronal artificial por Frank Rosenblatt en 1957; ELIZA, desarrollada en el Massachussets Institute of Technology (MIT) por Joseph Wizenbaun, en 1966, programa que actuaba como psicoanalizador y primer *chatbot* que incorporaba procesamiento del lenguaje natural humano para que las computadoras se comunicaran con el hombre. En 1969, Marvin Minsky, al que suele atribuírsele el término IA, escribe *Perceptrones*, el trabajo fundamental del análisis de las redes neuronales artificiales. En 1997 la supercomputadora Deep Blue*,* creada en *IBM,* vence en una partida de ajedrez al campeón del mundo Garri Kaspárov (Ponce Cruz, 2010). En los últimos años, gracias a la expansión de Internet, ha crecido rápidamente el interés de desarrollar *softbots* (robots *software*) que deambulan por Internet para encontrar información que consideran útil para sus usuarios. Han sido muchas las teorías, trabajos y desarrollos que han conducido hasta lo que es hoy en día la IA; la filosofía, matemáticas, economía, neurociencia, psicología, ingeniería computacional, teoría de control y cibernética y lingüística son algunas de las ramas del conocimiento que han aportado al desarrollo de la IA (Russell y Norvig, 2008).

Inteligencia artificial e Internet de las cosas

En la actualidad, dos importantes plataformas se han combinado para mejorar las interacciones humano-máquina y mejorar la gestión y análisis de datos (VIU, 2021): La inteligencia artificial (IA) y el Internet de las cosas (siglas en inglés *IoT* de *Internet of things*). Para entender cómo la inteligencia artificial e Internet de las cosas se integran, es necesario tener un conocimiento general sobre cada uno de estos términos.

Inteligencia artificial

La inteligencia artificial y el Internet de las cosas son temas novedosos. La increíble cantidad de dispositivos y la gran cantidad de información requiere de herramientas para analizar la información y obtener datos importantes de ella. El término IoT fue acuñado por Kevin Ashton en 1999. En aquel entonces la mayor parte de la información que procesaban las computadoras era entregado a estas por sus usuarios. Ashton propuso que las computadoras obtendrían información ellas mismas sin intervención de usuario alguno. Esto ya es una realidad, una gran cantidad de dispositivos tales como sensores, actuadores y teléfonos inteligentes se encuentran conectados al Internet. Las personas con dispositivos de vestir, los animales portando RFDIs, y las cosas de nuestro día a día —refrigerador, horno, electrodomésticos en general—, dentro y fuera de casa se convierten en proveedores de información (Vega-Luna et al., 2021).

Independientemente de las tecnologías que se usen para conectar los dispositivos, sin duda la combinación de los conceptos IA e IoT, son ya una revolución, prácticamente en cualquier ámbito: personal, hogar, calles, edificios, incluso en el espacio exterior.

Ramas de la IA

Dentro de la IA pueden encontrarse distintas ramas, algunas de ellas se describen a continuación:

- Aprendizaje máquina (*machine learning, ML*). Es un subcampo de la IA, se relaciona con asociaciones de aprendizaje complejo usando datos. En otras palabras, son algoritmos que aprenden de los mismos datos sin necesidad de programarse en un lenguaje de computadora explícitamente (Gutiérrez Pulido et al., 2018). El ML diseña y estudia herramientas informáticas que utilizan la experiencia para tomar decisiones futuras. Su objetivo fundamental es generalizar o inducir una regla desconocida a partir de ejemplos donde esa regla es aplicada. Tiene aplicaciones muy diversas, entre las cuales están: motores de búsqueda, visión por computadora (procesar, analizar y reconocer información),

reconocimiento de textos en manuscrito, diagnósticos médicos, reconocimiento de voz, detección de fraude y prevención en el sector bancario (en tarjetas de crédito por ejemplo), sector financiero (análisis del mercado de valores), clasificación de secuencias de ADN, juegos, robótica, para COVID-19 (diagnóstico de pacientes, población de mayor riesgo de contagio, predicción de la dispersión del virus). Hoy en día, existen diversas empresas (Google, Microsoft, Amazon e IBM, por ejemplo) que ofrecen servicios en la nube basados en aprendizaje máquina, pues ofrecen herramientas y técnicas para predecir o clasificar eventos sobre un conjunto de interacciones entre variables de una base de datos en particular (Bisong, 2019; Sharma et al., 2021). Dentro de los tipos de aprendizaje que pueden encontrarse en Aprendizaje Máquina están:

a) Aprendizaje supervisado: el algoritmo utiliza datos/variables etiquetados u organizados previamente para indicar cómo debe categorizarse la información, o bien, sobre lo que se quiere predecir (respuestas correctas). En este tipo de aprendizaje se requiere de la intervención del humano para dar retroalimentación, es decir, indicar qué está bien y qué está mal (Rouhiainen, 2018). Terminado el entrenamiento, se brindan nuevos datos sin etiquetas de las respuestas correctas y el algoritmo de aprendizaje predice un resultado. Ejemplos de algoritmos que usan este tipo de entrenamiento son: Regresión lineal, Regresión logística, *Support Vector Machine* (Máquinas de Vectores de Soporte) y Árboles de decisión, *Random Forest* (Sarker, 2021).
b) Aprendizaje no supervisado: el algoritmo se entrena usando un conjunto de datos sin etiqueta, es decir, nunca se le indica al algoritmo las clases de salida esperadas, esto con la finalidad de que el algoritmo encuentre por sí solo patrones o algún tipo de organización que simplifique el análisis, por lo

que es de carácter exploratorio con la intención de ayudar a entender el conjunto de datos. Se prefiere principalmente para la reducción de características, así como para agrupar (Sharma et al., 2021).

c) Aprendizaje por refuerzo: este algoritmo permite a los agentes inteligentes y máquinas evaluar automáticamente el comportamiento óptimo en un contexto o ambiente particular para mejorar su eficiencia. Aprende de los errores, con base a ensayo-error, pues su entrada es la retroalimentación que obtiene del mundo exterior como respuesta a sus acciones. El algoritmo aprende por sí solo a partir de recompensas o penalizaciones, y su meta es maximizar la recompensa y minimizar riesgos (Sarker, 2021).

- *Aprendizaje Profundo (AP).* En inglés, *Deep Learning (DL)*, subcategoría de aprendizaje máquina, se refiere a un aprendizaje basado en algoritmos que utilizan estructuras profundas de capas de unidades de proceso (neuronas artificiales) para encontrar patrones en los datos. A través de estas redes neuronales, AP puede realizar tareas complejas de aprendizaje que son difíciles de realizar por una computadora. Proporciona un aprendizaje automático de características y su representación es jerárquica, en varios niveles. Una red neuronal en AP puede tener cientos de capas ocultas, cada una con varias neuronas como unidades de procesamiento. El AP representa un acercamiento al modo de funcionamiento del sistema nervioso humano, por lo que se considera una tecnología clave en el futuro de la inteligencia artificial. Las capas iniciales llevan a cabo un procesamiento simple de los datos de entrada para aprender las características fáciles, y esa salida va a capas superiores que se encargan de realizar un aprendizaje de características más complejas. Entre algunas aplicaciones en las cuales han tenido éxito pueden mencionarse: reconocimiento de imágenes, agricultura, educación, cambio climático, comprensión del lenguaje

en sus variados significados contextuales (Díaz-Ramírez, 2021; Bisong, 2019; Sharma et al., 2021).

- Métodos probabilísticos. Estos algoritmos también entrenan y predicen, y utilizan redes Bayesianas o cadenas de Markov para tratar con la incertidumbre de todo proceso de aprendizaje, esto al tener que lidiar con información que tiene ruido, tiene pocos datos, está incompleta o que a veces es contradictoria (Pérez, 2019).
- Computación evolutiva y algoritmos genéticos. Estos algoritmos abordan la resolución automática de problemas de automatización y, a través de un proceso de búsqueda heurístico, simulan la selección natural siguiendo un proceso de evolución de individuos a través de acciones aleatorias. Estas simulaciones se realizan en una computadora a través de operaciones de selección, mutación y recombinación, abordan la resolución automática de problemas de automatización. Estos mecanismos se simulan en una computadora realizándo operaciones de selección de supervivientes, mutación y recombinación (Carmona Suárez y Fernández Galán, 2020).
- Visión por computadora. Este campo permite que las computadoras aprendan a reconocer una imagen y sus características, con el propósito de que las máquinas entiendan el mundo (González García et al., 2018).

Tipos de inteligencia artificial

A partir de algoritmos inteligentes, puede dotarse a las máquinas de cuatro formas diferentes de inteligencia artificial, las cuales se describen a continuación de manera general (Brianza, 2019; Harris, 2022).

1. Máquinas reactivas. Son las que tienen la forma más básica de IA, pues solo son capaces de usar inteligencia para percibir y reaccionar a los "desafíos" que el mundo les presenta. No pueden almacenar información por lo que no tiene memoria de experiencias pasadas para realizar toma de decisiones en tiempo real. Siempre re-

accionarán de la misma manera cada vez ante el mismo estímulo, por lo que se diseñan para realizar tareas especializadas. Un ejemplo de máquina reactiva es *Deep Blue* (supercomputadora para jugar ajedrez). Este tipo de máquinas son confiables para realizar tareas repetitivas.

2. Máquinas con memoria limitada. La IA con memoria limitada puede almacenar datos previos y predecir. Esto es posible porque tiene programado un conjunto posible de respuestas a problemas y, además, recopila información del ambiente que retiene por corto tiempo para después hacer una evaluación y tomar decisiones. En esencia, busca en el pasado pistas para determinar lo que viene después. Cuando se utiliza IA con memoria limitada, deben seguirse seis pasos: 1) creación de datos de entrenamiento, 2) definición del modelo de aprendizaje máquina, 3) el modelo debe ser capaz de predecir, 4) el modelo debe ser capaz de recibir retroalimentación humana o ambiental, 5) la retroalimentación debe almacenarse como datos y 6) estos pasos deben reiterarse como un ciclo. Existen tres principales modelos de aprendizaje máquina que utilizan IA con memoria limitada:
   - Aprendizaje reforzado, a través del cual aprende a realizar mejores predicciones a través de repetición de pruebas y errores.
   - Red de memoria a corto plazo (LSTM, siglas en inglés de *Long short term memory*), que utiliza datos del pasado para predecir el siguiente elemento de una secuencia. Considera la información más reciente como la más importante al momento de hacer las predicciones y disminuye los datos más antiguos, aunque los sigue utilizando para hacer conclusiones.
   - Redes adversarias generativas evolutivas (E-GAN, siglas en inglés de *Evolutionary Generative Adversarial Networks*), que evolucionan en el tiempo, creciendo para explorar rutas ligeramente modificadas basadas en experiencias anteriores con cada nueva decisión. Este modelo constantemente busca una

mejor ruta y utiliza simulaciones y estadísticas, o del azar, para predecir resultados a través de su ciclo de mutación evolutiva.

3. Máquinas con una teoría de la mente. Esta es solo teórica, puesto que no se cuenta todavía con las capacidades tecnológicas y científicas para alcanzar este nivel de IA. Se basa en la premisa psicológica de entender y predecir comportamientos de otros individuos a través de la comprensión de pensamientos, ánimo, ideas y emociones. Esto significa que la IA entendería cómo se sienten los humanos, los animales y otras máquinas, y así tomar decisiones a través de la autorreflexión y determinación para utilizar esta información y tomar sus propias decisiones.
4. Máquinas con conciencia propia. Una vez lograda la teoría de la mente, habría la posibilidad, en el futuro, de que la IA administre emociones externas y se vuelva consciente de sí misma. Esto implica que la IA tuviera conciencia a nivel humano y comprendiera su existencia en el mundo, así como la presencia y el estado emocional de los demás. Para esto se requiere que los investigadores humanos comprendan la premisa de la conciencia y luego aprendan a replicarla para integrarla a las máquinas.

Internet de las cosas

La idea del Internet de las cosas (*IoT*) y los dispositivos inteligentes surgió a principios de la década de 1980, su objetivo era permitir a los dispositivos comunicarse unos con otros directamente de una manera de igual a igual. En la actualidad, IoT está transformando la forma en que interactuamos con los dispositivos que tenemos en casa, en el trabajo y ciudades, por ello se considera uno de los pilares clave de la cuarta revolución industrial. A través de esta red de dispositivos se recolecta gran cantidad de datos de un sinnúmero de actividades diversas, brindando así a las tecnologías basadas en IoT una nueva perspectiva sobre el progreso de campos como: ingeniería, agricultura, industria, transporte, medicina, construc-

ción de casas inteligentes, administración de energía (energía inteligente), dispositivos vestibles (*wearable*), manejo de desperdicios, entre otros (Nižeti- et al., 2020).

Por lo mencionado anteriormente, es entendible el por qué la firma de investigación de mercado IoT Analytics predijo que para el 2025 existirán 21.5 billones de dispositivos IoT. Su crecimiento ha sido exponencial a tal grado que incluso, solos y/o juntos, los dispositivos IoT ofrecen nuevas funciones y capacidades para empresas y consumidores; por ejemplo, a través de plataformas como Amazon Echo/Alexa, Google Home o Apple HomeKit, se pueden automatizar funcionalidades en una casa, por ejemplo, administración inteligente de energía, automatización de aires acondicionados, activación de alarmas y cámaras de seguridad, detectores de filtraciones de agua, control de termostatos, solo por mencionar algunos. Internet de las cosas presenta oportunidades para utilizar datos de manera novedosa mediante el uso de redes sociales, datos de geolocalización e incluso, análisis y aprendizaje automático (AWS, 2021a; Greengard, 2021).

Como se menciona anteriormente, las aplicaciones de IoT las encontraremos cada vez en mayor número en cualquier ámbito: personal, hogar, calles, edificios, ciudades, industria y manufactura, incluso los destinos turísticos. Quizás el entorno con más crecimiento es la industria y la manufactura, debido al potencial tan importante del número de dispositivos conectados y los diferentes beneficios que se obtienen: soluciones de energía verde, automatización de fábricas, ubicación de activos, y optimización e incremento de la producción.

Aunque en la última década, los microprocesadores, el cómputo en la nube, la inteligencia artificial y las comunicaciones han crecido a tal grado que han dado una base firme para IoT (Greengard, 2021), estas tres tecnologías se consideran clave para IoT:

1. Inteligencia artificial, pues utiliza técnicas de aprendizaje automático para desarrollar contramedidas a varios tipos de ataques de *IoT* (Zaman et al., 2021).
2. Redes móviles de 5ta. generación, que proporcionan tecnologías requeridas para el despliegue ubicuo de la tecnología IoT, pues al ser extremadamente rápidas,

permiten el procesamiento de datos en tiempo real de los dispositivos IoT, haciendo realidad desarrollos como: carros autónomos, ambientes inteligentes, transporte inteligente y e-salud (Shafique et al., 2020).

3. *Big Data*, que permite realizar análisis de grandes cantidades de datos para extraer información, conocimiento y hacer inferencias o predicciones proveniente de dispositivos de IoT (Li et al., 2021).

Tendencias de la IA aplicada al IoT

La clave para que el Internet de las cosas en conjunto con inteligencia artificial y las plataformas de IoT sean realmente potentes y mejoren su rendimiento operativo, es que las cargas de trabajo de inteligencia artificial (IA) y de aprendizaje automático o máquina (ML) sean lo más equilibradas posible. A pesar de la enorme cantidad de datos de IoT capturados, las organizaciones no alcanzan sus objetivos de gestión del rendimiento empresarial porque la IA y el ML no se adaptan a los desafíos en tiempo real que estas enfrentan. Estos retos se describen en el siguiente apartado.

Desafíos de crecimiento del IoT con IA

Muchas organizaciones están buscando iniciativas basadas en inteligencia artificial de vanguardia, para convertir los datos de monitoreo de procesos y producción en tiempo real de IoT en resultados más eficientes. Con ello, las empresas deben enfrentar los siguientes desafíos (BigData, 2021):

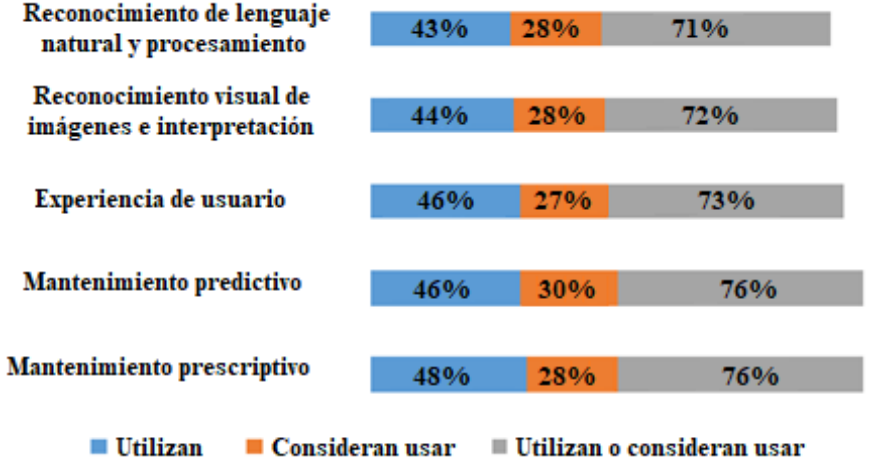
1. Los dispositivos de punto final de I-IoT (siglas en inglés de *Industrial Internet of things*, Internet industrial de las cosas) e IoT deben progresar más allá del monitoreo en tiempo real para proporcionar inteligencia contextual como parte de una red. Los equipos de operaciones y TI (Tecnologías de Información) empresarial requieren puntos finales más inteligentes contextualmente para mejorar la visibilidad de un extremo a otro en las redes basadas en sensores de IoT en tiempo real. Los planes de desar-

rollo incluyen tener sistemas basados en IA que brinden recomendaciones de mejora del rendimiento función de los resultados del modelo de aprendizaje automático.

2. El modelado de IA y ML debe ser fundamental para una arquitectura I-IoT/IoT, y no un complemento. El objetivo es admitir el procesamiento de modelos en múltiples etapas de una arquitectura I-IoT / IoT mientras se reduce el rendimiento y la latencia de la red.
3. Los dispositivos I-IoT/IoT deben tener un diseño lo suficientemente adaptable para admitir actualizaciones de algoritmos. La propagación de algoritmos a través de una red I-IoT/IoT al nivel del dispositivo es esencial para que toda una red logre y mantenga la sincronización en tiempo real. La lógica de control estadístico de procesos integrada en los dispositivos I-IoT proporciona datos de productos y procesos en tiempo real integrales para el éxito de la gestión de la calidad.
4. Las plataformas I-IoT/IoT con un enfoque de mercado único y diferenciado. Para que una plataforma de este tipo gane escalamiento, cada uno debe especializarse en un mercado vertical determinado, así como proporcionar las aplicaciones y herramientas para medir, analizar y ejecutar operaciones complejas.
5. La transferencia de conocimiento debe ocurrir a escala. A medida que los trabajadores se retiran y las organizaciones abandonan el modelo tradicional de aprendizaje, la transferencia de conocimientos se convierte en una prioridad estratégica. El objetivo es equipar a la última generación de trabajadores con dispositivos móviles que sean lo suficientemente inteligentes contextualmente para proporcionar datos en tiempo real sobre las condiciones actuales al mismo tiempo que brindan inteligencia contextual y conocimiento histórico. Combinar el conocimiento obtenido de los trabajadores que se jubilan con técnicas de IA y ML para ayudar en la capacitación de los trabajadores actuales y futuros es clave.

Los datos de IoT pueden impulsar mejoras en el rendimiento, dependiendo de la toma de decisiones de las empresas. En una encuesta de Microsoft se encontró que el 79% de las empresas que adoptan IoT consideran a la IA como un componente central o secundario de sus estrategias, como se aprecia en la figura 1.

Figura 1. Razones para la adopción en las empresas de IA en IoT



Fuente: Elaboración propia, adaptado de *BigData* (2021).

De acuerdo con un análisis de los casos de uso proporcionados en el estudio *Microsoft IoT Signals Edition* 2 (Columbus, 2020) y las intervenciones que Venture Beat ha tenido con los líderes de fabricación, cadena de suministro y logística, se obtuvieron las siguientes recomendaciones que pueden mejorar el rendimiento de IoT (BigData, 2021):

- Incluir ganancias de ingresos y reducciones de costos en los casos de negocios de la empresa. Cuantas más redes I-IoT/IoT entregue la plataforma de datos para respaldar la generación de informes y análisis en tiempo real de la gestión del rendimiento empresarial, es más probable que sea más exitosa.
- Diseñar arquitecturas I-IoT/IoT hoy para la expansión de dispositivos de uso de IA en el futuro. El futuro de

las redes I-IoT/IoT estará dominado por dispositivos de punto final capaces de modificar algoritmos al tiempo que imponen el acceso con menos privilegios. La creciente inteligencia de los sensores y las mejoras en el monitoreo de procesos en tiempo real los están convirtiendo en un vector de amenaza principal en las redes. El diseño en microsegmentación y la aplicación del acceso menos privilegiado al sensor individual se están logrando hoy en día en todos los sitios de fabricación inteligente.

- Planificar modelos de IA que puedan escalar a contabilidad y finanzas desde las operaciones. Saber cómo las compensaciones en los proveedores, la selección de maquinaria y las asignaciones de la tripulación afectan las tasas de rendimiento y las ganancias de productividad es clave. Hacer concesiones inmediatas en el análisis de la calidad del producto ayuda a aliviar las variaciones en los costos reales de cada proyecto, gracias a los datos de I-IoT.
- Considerar la formación de modelos de aprendizaje automático a nivel de algoritmo de dispositivo desde el principio. El objetivo es saber cómo y dónde corregir el rumbo en un proceso determinado basándose en el análisis de datos en tiempo real.

El aprendizaje automático usa algoritmos estadísticos que aprenden a partir de los datos existentes (proceso llamado entrenamiento), a fin de tomar decisiones acerca de datos nuevos (proceso llamado inferencia). Durante el entrenamiento, se identifican los patrones y las relaciones en los datos a fin de crear un modelo. El modelo permite a un sistema tomar decisiones inteligentes en relación con datos que no encontró anteriormente. Al optimizar los modelos se comprime el tamaño del modelo para que se ejecute rápidamente (AWS, 2021b).

La capacitación y optimización de modelos de *machine learning* requiere grandes cantidades de recursos de cómputo, por lo que este proceso resulta adecuado ejecutarse en la nube. Sin embargo, la inferencia necesita una capacidad de procesamiento mucho menor y con frecuencia se realiza en tiempo real cuando se encuentran disponibles datos nuevos. La obtención de resultados

de inferencia con un nivel de latencia muy bajo es importante para garantizar que sus aplicaciones con IoT puedan responder rápidamente ante eventos locales. AWS IoT Greengrass facilita realizar las inferencias de aprendizaje automático localmente en los dispositivos con modelos que están creados, entrenados y optimizados en la nube (AWS, 2021b).

Entre los principales beneficios destacan: la flexibilidad, la implementación de modelos en los dispositivos conectados con unos pocos clics, el rendimiento de inferencia acelerado, la fácil ejecución de inferencia en más dispositivos, la creación de modelos más precisos. En AWS se presentan casos de uso para mantenimiento industrial predictivo, agricultura de precisión, seguridad, sector minorista y hotelería, y procesamiento de videos (AWS, 2021b).

Cómputo en la nube para IoT con IA

La nueva ola del *IoT* se enfoca en potencializar la analítica de los datos generados aprovechando los recursos y herramientas del *Big Data*. El valor real del IoT va más allá de la infraestructura de *hardware* utilizada para el monitoreo en tiempo real, se extiende a todos los servicios de valor añadido que se pueden generar con la utilización e integración del aprendizaje automático y la inteligencia artificial.

De acuerdo con el trabajo de investigación *IoT data-enabled services: value chain, companies to watch, and cloud wars* de la consultora ABI Research, se estima que los servicios de ML e IA en el ámbito del IoT crecerán 40% anualmente y alcanzarán un valor estimado en 3,600 millones de dólares para el año 2026 (ABI Research, 2021).

Como ya es algo conocido, la situación de pandemia por COVID-19 afectó negativamente a muchas industrias y sus efectos negativos tardarán tiempo en eliminarse. Sin embargo, como lo señala ABI Research (2021), una de las industrias menos afectadas fue la relacionada con el IoT, incluso se generaron nuevas áreas de oportunidad para la creación de industrias innovadoras, esto como consecuencia a la migración del trabajo presencial al trabajo a distancia. Particularmente el área de la analítica del IoT creció aceleradamente y sigue en constante avance. Ahora las industrias requieren integrar herramientas de ML e IA para realizar el monitoreo permanente y en tiempo

real del trabajo en casa, así como la visualización y administración de los activos de la empresa, incluso el apoyo remoto a los empleados y el mantenimiento predictivo son procesos altamente demandados actualmente. En esas nuevas necesidades de la industria es donde el ML e IA tienen una aplicación fundamental para obtener información útil y de calidad, de la gran cantidad de datos que se generan por los diferentes actores e involucrados en una industria con trabajo a distancia.

Para hacer frente a estas nuevas necesidades, los principales proveedores de servicios en la nube cuentan con recursos de plataformas como servicios (PaaS) y *software* como servicios (SaaS) que las industrias pueden aprovechar para implementar herramientas de ML e IA que utilicen los datos generados por la IoT y generen información en tiempo real para el apoyo en la toma de decisiones.

En los siguientes párrafos se presenta una de las herramientas y servicios para la implementación de ML e IA en el IoT que ofrece Microsoft que es uno de los principales proveedores de servicios en la nube. El objetivo de esta información es brindar un panorama general de las opciones y la manera de cómo beneficiarse del *Big Data* y el cómputo para potencializar la analítica del IoT con herramientas de ML e IA.

La empresa Microsoft con su plataforma Azure ofrece una amplia variedad de servicios administrados y de plataforma que abarcan el cómputo de perímetro y la nube que permiten conectar, supervisar, administrar y controlar miles de millones de recursos de IoT. También incluye seguridad y sistemas operativos para dispositivos y equipos, junto con datos y análisis que ayudan a las empresas a crear, implementar y administrar aplicaciones de IoT. Desde la visión de la empresa, el IoT se integra por tres componentes que deben trabajar juntos y administrarse por los servicios que ofrece Azure: 1) las cosas, que son todos los objetos físicos que se conectan a la nube de forma persistente o intermitente, 2) los conocimientos, definidos como la información recopilada por las cosas, que es analizada y convertida en conocimiento útil por el personal o mediante inteligencia artificial y 3) las acciones, forma en la que las personas responden a esa información y la conectan a sus negocios, así como a los sistemas y herramientas que utilizan. Bajo estas premisas, Microsoft Azure ofrece productos y servicios que se presentan en la tabla 1.

Tabla 1. Productos y servicios que ofrece Microsoft Azure

Azure IoT Hub (https://azure.microsoft.com/es-mx/services/iot-hub/)	Permite habilitar comunicación segura y confiable entre la aplicación de Internet de las cosas y los dispositivos que administra. Azure IoT Hub ofrece un *back-end* de solución hospedado en la nube que permite conectarse prácticamente a cualquier dispositivo. Implementa autenticación por dispositivo y una administración de dispositivos integrada.
Azure IoT Central (https://azure.microsoft.com/es-mx/services/iot-central/).	Acelera la creación de soluciones de IoT, reduce la carga, el costo de desarrollo y las operaciones de administración de IoT.
Azure Digital Twins (https://azure.microsoft.com/es-mx/services/digital-twins/)	Permite crear una representación digital de cosas, lugares, procesos empresariales y personas reales. Obtenerse información que permita mejorar los productos y optimizar las operaciones y los costos, y crear experiencias de cliente realmente innovadoras.
Azure IoT Edge (https://azure.microsoft.com/es-mx/services/iot-edge/)	Permite ampliar la inteligencia y los análisis de la nube moviendo las cargas de trabajo y la lógica de negocios de la nube a los dispositivos perimetrales.
Azure RTOS (https://azure.microsoft.com/es-mx/services/rtos/).	Simplifica el desarrollo y la conectividad de IoT con un sistema operativo pequeño pero eficaz que proporciona un rendimiento ultrarrápido y confiable para los dispositivos con recursos limitados.
Azure Machine Learning (https://docs.microsoft.com/es-mx/azure/machine-learning/overview-what-is-azure-machine-learning).	Permite acelerar y administrar el ciclo de vida de los proyectos de aprendizaje automático. Los profesionales de aprendizaje automático, científicos de datos e ingenieros pueden usarlo en sus flujos de trabajo diarios: entrenamiento e implementación de modelos y administración de MLOps.

Hoy en día somos testigos de cómo la inteligencia artificial contribuye al avance del Internet de las cosas en áreas como descubrimiento de información, preparación y visualización de datos, análisis predictivos y avanzados de información obtenida por la gran cantidad de dispositivos conectados. Es evidente cómo el cómputo en la nube, *machine learning, Big Data* e Internet permiten el almacenamiento y procesamiento de gran cantidad de información, que puede distribuirse y obtenerse gracias a la tecnología 5G. Por ejemplo, recientemente se tiene también el desarrollo de gemelos digitales, tecnología de *software* que tiene como objetivo representar digitalmente una entidad física, objetos físicos o plantas virtuales y que permitirían a las industrias observar su comportamiento sin poner el riesgo a los originales y con costos muy bajos. De nuevo, se requiere la conjugación de *Big Data*, IA, *Cloud Computing,* IoT y ML, para mejorar procesos que van desde el control hasta la optimización de estos, y tratar aspectos de sostenibilidad, innovación inteligente, salud y seguridad.

Conclusiones

El IoT e IA han dado pie al surgimiento de una diversidad de nuevos objetos inteligentes con capacidad de detección, procesamiento, gestión, interactividad, aprendizaje y comunicación. Dado que los procesos realizados con IoT generan gran cantidad de datos, a través de la IA puede procesarse dicha información y darle sentido. Es por ello que reconocidos especialistas predicen una completa interacción entre humanos y máquinas para el 2030, porque junto con *Big Data, Data Mining, Cloud Computing*, robótica y automatización actual, la AIoT hará posible una integración completa entre el mundo físico y el virtual. Se requerirá aprovechar la inmensa información de un sinnúmero de dispositivos inteligentes que, a través de un procesamiento adecuado, permitirán contar con entornos que busquen mejorar la calidad de vida y entre los cuales podemos mencionar: tecnología de vestir (*Wearable*), realidad virtual, inmuebles inteligentes, ciudades inteligentes, industrias inteligentes y entornos empresariales.

Referencias

ABI Research. (2021). *IoT data-enabled services: value chain, companies to watch, and cloud wars*. https://www.abiresearch.com/market-research/product/7778820-iot-data-enabled-services-value-chain-comp/

AWS (2021a). *AWS IoT para los hogares conectados. Amazon Web Services*. https://aws.amazon.com/es/iot/solutions/connected-home/

AWS (2021b). Inferencia de ML de AWS IoT Greengrass. Implementación de modelos de machine learning optimizados para ejecutarse en dispositivos AWS IoT Greengrass. https://aws.amazon.com/es/greengrass/ml/

BigData (2021). *Descubriendo el potencial de IoT con inteligencia artificial y aprendizaje automático. Maching Learning*. https://topbigdata.es/descubriendo-el-potencial-de-iot-con-inteligencia-artificial-y-aprendizaje-automatico/

Bisong, E. (2019). *Building machine learning and deep learning models on Google Cloud Platform. A comprehensive guide for beginners.* Apress Berkeley. https://doi.org/10.1007/978-1-4842-4470-8

Brianza, A. (2019). Audiovisión e Inteligencia Artificial. En A. Gómez, A. León Grisales, O. E. Tamayo y S. H. Sierra (Eds.), *Diseño y Creación* (Primera ed., pp. 177-181). Universidad de Caldas.

Carmona Suárez, E. J. y Fernández Galán, S. (2020). *Fundamentos de computación evolutiva.* Marcombo.

Columbus, L. (2020, 12 de octubre). 10 insights from Microsoft's latest IoT signals report. *Forbes.com.* https://www.forbes.com/sites/louiscolumbus/2020/10/12/10-insights-from-microsofts-latest-iot-signals-report/?sh=510017745fa5

D'Arc, T. (2020). *Qué es la Inteligencia Artificial: 8 ejemplos en tu vida diaria*. https://www.smarthint.co/es/o-que-e-inteligencia-artificial-exemplos/?utm_source=blog&utm_medium=post&utm_campaign=buscainteligente

Díaz-Ramírez, J. (2021). Aprendizaje automático y aprendizaje profundo. *Ingeniare. Revista chilena de ingeniería, 29*(2). http://dx.doi.org/10.4067/S0718-33052021000200180

Dick, S. (2019). Artificial Intelligence. *Harvard Data Science Review*. https://doi.org/10.1162/99608f92.92fe150c

Ghosh, I. (2020). *AIoT: When artificial intelligence meets the Internet of things*. https://www.visualcapitalist.com/aiot-when-ai-meets-iot-technology/

González García, C., Núñez-Valdez, E. R., García-Díaz, V., Pelayo G-Bustelo, B. C. y Cueva Lovelle, J. M. (2018). A review of artificial intelligence in the Internet of things. *International Journal of Interactive Multimedia and Artificial Intelligence, 5*(4), pp. 9-20. https://digibuo.uniovi.es/dspace/bitstream/handle/10651/52310/A%20Review.pdf?sequence=1

Greengard, S. (2021). *The Internet of Things.* The MIT Press.

Gutiérrez Pulido, J. R., Ramos Michel, E. M. y Acosta Díaz, R. (2018). Inteligencia artificial y aprendizaje máquina: aplicaciones y tendencias. En A. Román Gallardo, S. Sandoval Carrillo, M. E. Cabello Espinosa y J. R. Herrera Moralesy (Eds.), *Tecnologías disruptivas de información.* Universidad de Colima. http://ww.ucol.mx/content/publicacionesenlinea/adjuntos/Tecnologias-disruptivas-de-informacion_465.pdf

Harris, D. (2022). *Artificial Intelligence. What is Artificial Intelligence? How does AI work?* https://builtin.com/artificial-intelligence

Li, W., Chai, Y., Khan, F., Jan, S. U., Verma, S., Menon, V. G. y Li, X. (2021). A comprehensive survey on machine learning-based Big Data analytics for IoT-Enabled smart healthcare system. *Mobile Networks and Applications,* 26, pp. 234-252. https://doi.org/10.1007/s11036-020-01700-6

Newell, A. y Simon, H. (1956). The Logic Theory Machine - A complex information processing system. *IRE Transactions on Information Theory, (2)*3, pp. 61-79. https://doi.org/10.1109/TIT.1956.1056797

Nižeti-, S., Šoli-, P., López-de-Ipiña González-de-Artaza, D. y Patrono, L. (2020). Internet of Things (IoT): opportunities, issues and challenges towards a smart and sustainable future. *Journal of Cleaner Production, 274*. https://doi.org/10.1016/j.jclepro.2020.122877

Pérez, F. A. (2019). *El enfoque probabilístico en Inteligencia Artificial.* (GESI, Ed.) http://www.ugr.es/~gesi/analisis-gesi-14-2019.pdf

Ponce Cruz, P. (2010). *Inteligencia Artificial con aplicaciones a la Ingeniería* (Primera ed.). Alfaomega.

Rouhiainen, L. (2018). *Inteligencia Artificial. 101 cosas que debes saber hoy sobre nuestro futuro.* Alienta.

Russell, S. y Norvig, P. (2008). Inteligencia Artificial. Un enfoque moderno (Segunda ed.). Pearson Educación.

Sarker, I. H. (2021). Machine Learning: algorithms, real-world applications and research directions. *SN Computer Science, 2*(160).https://doi.org/10.1007/s42979-021-00592-x

Shafique, K., Khawaja, B. A., Sabir, F., Qazi, S. y Mustaqim, M. (2020). Internet of Things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios. *IEEE Access, 8*. https://doi.org/10.1109/ACCESS.2020.2970118

Sharma, N., Sharma, R. y Jindal, N. (2021). Machine learning and deep learning applications - a vision. *Global Transistions Proceedings, 2*, 24-28. doi:https://doi.org/10.1016/j.gltp.2021.01.004

Universidad Internacional de Valencia (VIU). (2021). *AIoT: un gran avance en Inteligencia Artificial y el Internet de las cosas*. Universidad Internacional de Valencia. https://www.universidadviu.com/co/actualidad/nuestros-expertos/aiot-un-gran-avance-en-inteligencia-artificial-y-el-internet-de-las

Vega-Luna, J., Lagos-Acosta, M., Salgado-Guzmán, G., Cosme-Aceves, J. y Tapia-Vargas, V. (2021). Control de sensores de presencia de equipo de cómputo usando Alexa e IoT. *Revista Ciencia, Ingeniería y Desarrollo Tec. Lerdo, (1)*7, pp. 13-18. http://revistacid.itslerdo.edu.mx/coninci2021/CID011.pdf

Zaman, S., Alhazmi, K., Aseeri, M. A., Ahmed, M. R., Khan, R. T., Kaiser, M. S. y Mahmud, M. (2021). Security threats and Artificial Intelligence based countermeasures for Internet of Things networks: a comprehensive survey. *IEEE Access, 9*. https://doi.org/10.1109/ACCESS.2021.3089681

Capítulo 8

Ejercicios prácticos de IoT

Armando Román Gallardo
José Román Herrera Morales
Sara Sandoval Carrillo

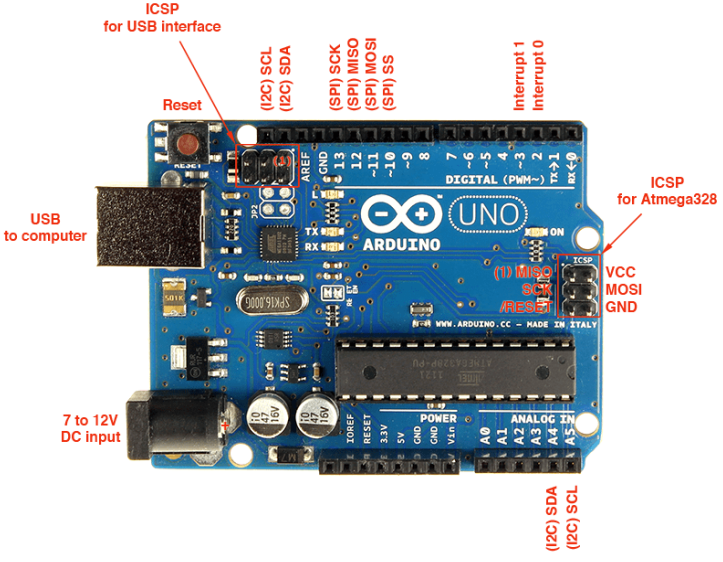
Resumen

Las ingenierías de software y de tecnologías de internet de la Facultad de Telemática promueven el desarrollo del Internet de las cosas de manera regional y nacional en México. Estas carreras se relacionan directamente con la necesidad de producir dispositivos electrónicos y aplicaciones de *software* que tengan la capacidad de conectarse y ser parte de las redes de telecomunicaciones que tienen sus aplicaciones en una gran variedad de áreas, como lo son las casas y edificios inteligentes, biomédicas, agricultura, carros conectados y, finalmente, lo que es la industria inteligente que pone a las empresas listas para la industria 4.0 o también llamado el Internet industrial de las cosas I-IoT (Boyes et al., 2018). Por lo anterior, hemos preparado una serie de ejercicios prácticos para que el lector, pueda interactuar con la electrónica, la conectividad, los protocolos de comunicación, el desarrollo en el nivel final para utilizarse por los usuarios o dentro de los dispositivos electrónicos de manera embebida, permitiéndose enviar y recibir desde los dispositivos hacia Internet y de Internet hacia los dispositivos, para censar información para la toma de decisiones o accionar otros dispositivos para realizar tareas específicas.

Práctica: conocer el entorno de desarrollo de Arduino, prendiendo y apagando un led

Arduino (figura 1) es una plataforma de creación de proyectos de electrónica de código abierto, esto quiere decir que cuenta con un *hardware* libre cuyos diagramas, así como sus especificaciones son públicas; de la misma manera, maneja el *software* libre, donde el código de los programas informáticos es accesible por cualquier persona, pudiendo utilizarlo y actualizarlo. Por lo anterior se convierte en una plataforma muy flexible que permite a otros fabricantes desarrollar sus propias placas y librerías. Además, se pueden crear diferentes microcomputadoras de una sola placa y, así, la comunidad que trabaja en estas, le puede dar diferentes usos y aplicaciones (Novillo-Vicuña et al., 2018).

Figura 1. Placa Arduino UNO



Fuente: Recuperada de https://i1.wp.com/www.gammon.com.au/images/ArduinoUno_R3_Pinouts.png.

Esta placa se basa en un microcontrolador ATMEL que es un circuito al cual se le pueden grabar instrucciones en el lenguaje de programación C, en el entorno llamado Arduino IDE, lo que te

permitirá interactuar con los circuitos de la placa para obtener un resultado específico. Arduino cuenta con una interfaz de entrada para conectar diferentes periféricos como sensores, teclados y hasta cámaras de video, entre otros, que trasladarán información hacia el microcontrolador y, una vez que esta se procesa, puede ser regresada a otros periféricos como pantallas, bocinas u otras placas mediante la interfaz de salida (Perkasa et al., 2021). Así pues, tenemos que Arduino es un proyecto y no solo un modelo específico, por lo que podrás encontrar muchas variaciones de este como las que se muestran en la figura 2.

Figura 2. Muestra de las diferentes versiones de placas de desarrollo que puedes encontrar del proyecto Arduino

Fuente: Recuperada de https://i.blogs.es/218ccc/formas-arduino/1366_2000.jpg

Objetivo general

Encender y apagar un LED con una placa Arduino UNO, ya sea para dejarlo fijo o de manera intermitente.

En esta práctica trabajarás la electrónica que permite encender o apagar un LED. Podemos considerar que los LED son actuadores porque solo se les puede cambiar el estado de prendido o apagado. Para lograr este resultado tendrás que programar en lenguaje C la aplicación en el Arduino IDE que puedes descargar de la siguiente página: https://www.arduino.cc/en/software para el sistema operativo que estés utilizando ya sea Windows, Linux o Mac OSX (Peña, 2020).

Materiales

1. LEDs, de distintos colores.
2. Una resistencia de 220Ω.
3. Una tarjeta Arduino Uno.
4. Un cable USB de impresora.
5. Una computadora.
6. Cables para el montaje del circuito (figura 3).
7. *Protoboard*.
8. Arduino IDE

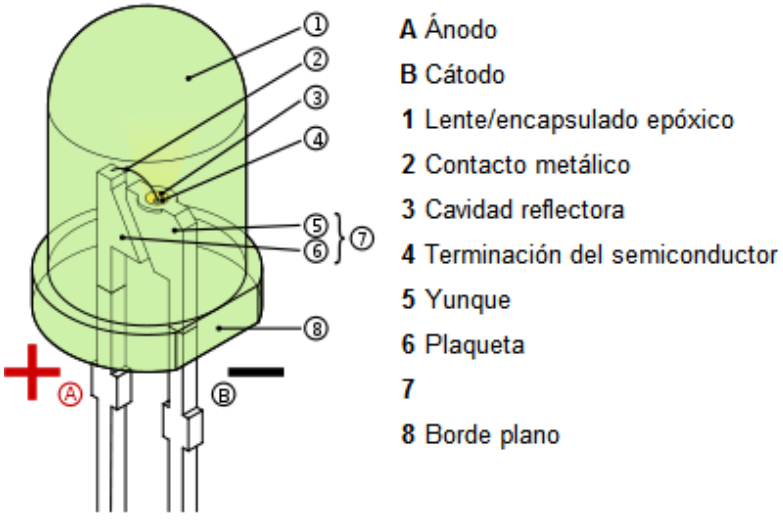
Figura 3. Cables dupont

Recuperado de https://natytec.com.mx/wp-content/uploads/2017/09/Icono12.jpg

Método para identificar la polaridad de un led

Un dispositivo diodo emisor de luz LED (figura 4) es utilizado como indicador o para iluminación en una gran cantidad de dispositivos, consume aproximadamente 2 voltios; originalmente solo emitía luz roja de baja intensidad, actualmente hay brillantes en el espectro infrarrojo visible y en el ultravioleta. La polaridad de un LED se puede conocer primero observando cuál es la pata más larga, esta siempre va a ser el ánodo y, por consiguiente, la otra será el cátodo, la que puedes identificar por la base del LED que tiene un borde plano y, finalmente, dentro del LED pues mirar que hay una plaqueta indicando el ánodo, mientras que un yunque señala el cátodo (Geek Factory, 2020).

Figura 4. Partes de un LED

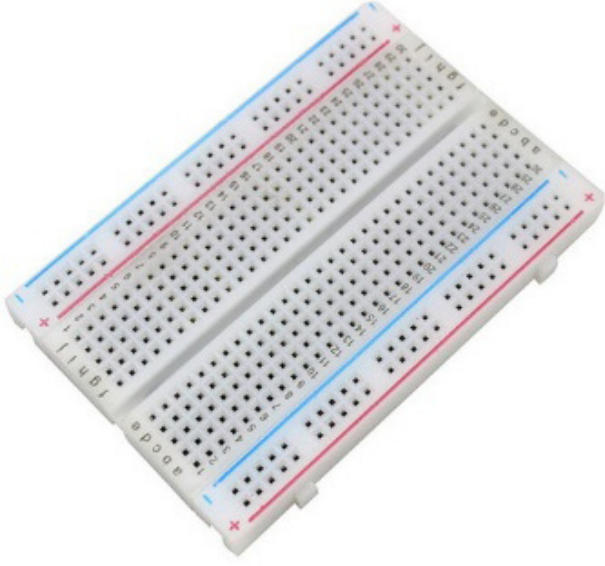


Fuente: Recuperado de https://www.ledtecnologia.com/wp-content/uploads/2016/09/led-caracteristicas.png

Protoboard

Para armar el circuito es necesario que utilices un *protoboard* (figura 5), el cual es un tablero formado por orificios en forma de líneas conectados eléctricamente entre sí; es en estos orificios donde se conectarán los componentes eléctricos y, de esta forma, poder crear prototipos de sistemas y dispositivos electrónicos que se pueden probar antes de que lleguen a imprimirse, para crear productos finales en la etapa de producción (Vergara, 2021).

Figura 5. *Protoboard*



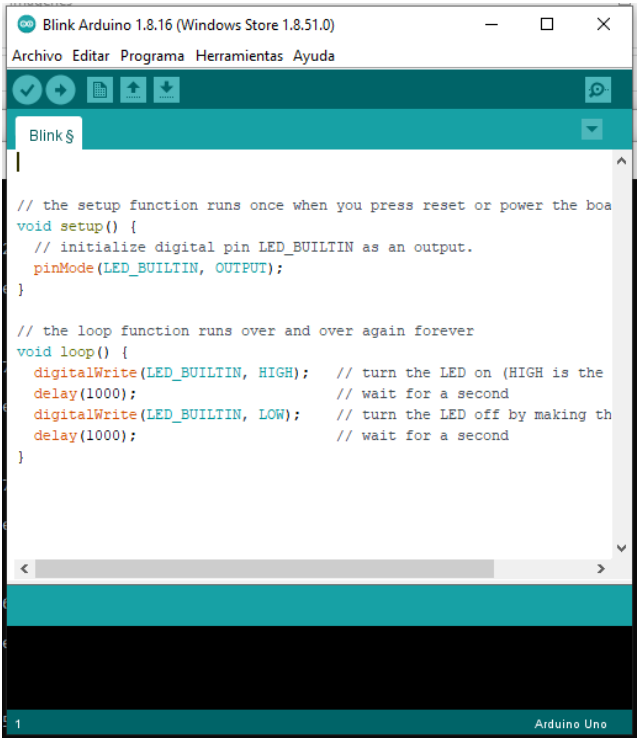
Fuente: Recuperado de https://http2.mlstatic.com/D_NQ_NP_949602-MLM43496119309_092020-O.jpg

Arduino IDE

Para que puedas empezar a programar la placa Arduino, necesitas descargar el programa Arduino IDE (del inglés *Integrated Development Environment*), el cual cuenta con las herramientas necesarias para escribir, editar, depurar y grabar en nuestra placa de Arduino, así como nuestros programas que son llamados *sketches*. Puedes

descargar este programa de la siguiente dirección https://arduino.cl/programacion/ (Peña, 2020).

Figura 6. Captura de pantalla del programa Arduino IDE



Fuente: Elaboración propia.

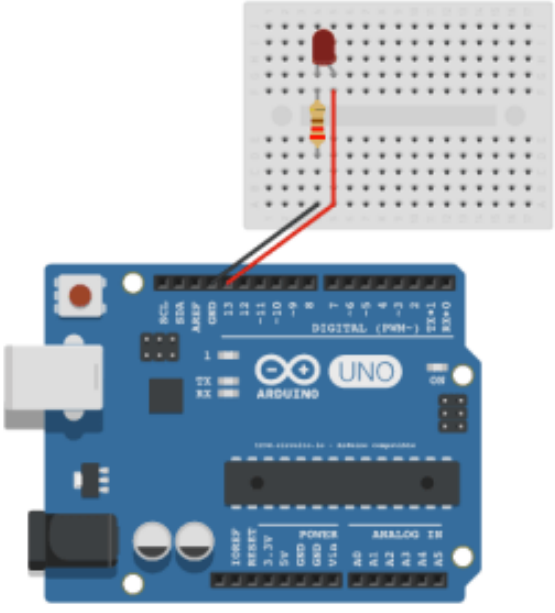
Requisitos previos

Es necesario contar con conocimientos previos de programación en lenguaje ANSI C. Si quieres conocer más acerca de la programación con Arduino puedes ver tutorial de programación de C para Arduino que se encuentra en YouTube, del usuario Tutoriales de ingeniería titulado *02 Arduino Desde Cero 2020 - Saltos de Línea(\n) - Repaso Lenguaje C [Capítulo 1*] que se encuentra en la dirección https://www.youtube.com/watch?v=8ArpMZ4Hm8Q.

Procedimiento

1. Armado del circuito electrónico. En la figura 5 se muestra cómo se debe armar el circuito en el que debes tener ya claro cuál es el ánodo y cátodo para poder realizar la conexión de este con la placa Arduino, donde debes conectar el ánodo al pin 13, y el cátodo seguido de la resistencia de 220Ω, conéctala al pin de tierra o también llamado GND (siglas del inglés *ground*).

Figura 5. Esquema de cómo debe montarse el circuito para poder prender y apagar el LED con la placa Arduino



Fuente: Recuperado de https://i2.wp.com/mecabot-ula.org/wp-content/uploads/practica1a.png?w=507

2. Antes de desarrollar el código en el IDE de Arduino tienes que realizar las siguientes acciones en el programa:
   a. Selecciona la tarjeta Arduino a través del menú en la opción de Herramientas -> placa.

b. Y después selecciona el puerto USB por el cual se va a conectar el Arduino Uno a la computadora.

3. Procedimiento para la codificación del programa:
   c. Crea un nuevo programa seleccionando Archivo->Nuevo.
   d. Escribe el código fuente del programa para encender y apagar el LED que se encuentra en el pin 13.

```
//Programa para encender y apagar un LED
//constante para manejar el pin donde estará conectado el LED
const int LED=13;

//sección de la configuración cuando arranque la ejecución
//del programa en la placa.
void setup() {
 //ponemos el pin en modo de salida
 pinMode(LED,OUTPUT);
}

void loop() {
 //Encendemos el LED poniendo el valor de salida en alto
 digitalWrite(LED,HIGH);
 //Hacemos una pausa de 1 segundos antes de apagarlo
 delay(1000);
 //Apagamos el LED poniendo el estado pin de salida en bajo.
 digitalWrite(LED,LOW);
 //Hacemos una pausa de 1 segundos antes de volver a encenderlo
 delay(1000);
}
```

4. Para guardar el programa en la computadora debes de seleccionar del menú Archivo->Salvar, asígnale el nombre al *sketch* o presiona las teclas Ctrl más la tecla S (Ctrl+S).
5. Ahora debes compilar el programa seleccionando del menú Programa->Verificar/Compilar o presionando las

teclas Ctrl más la tecla R (Ctrl + R), o del IDE el botón que tiene una flecha .

6. Si no hubo ningún problema al compilarlo, ya puedes subir tu programa a la placa Arduino UNO, y ver el resultado, para ello selecciona del menú Programa->Subir, otra opción es presiona las teclas Ctrl más la tecla U (Ctrl + U), o del IDE el botón que tiene una flecha hacia la derecha , después de ello podrás ver cómo enciende y se apaga el LED que conectaste en tu placa Arduino.

Práctica: conociendo el entorno de desarrollo del módulo NodeMCU 8266 utilizando Arduino IDE para programarlo

Objetivo general

Encender y apagar un LED utilizando el módulo NodeMCU ESP8266 mediante el protocolo HTTP. En esta práctica trabajarás la electrónica que permite encender o apagar un LED, en este caso será el que está interconstruido en la propia placa para fines prácticos. Podemos considerar que los LEDs son como los actuadores o accionadores que con ellos podríamos prender una luminaria, una bomba o abrir una electroválvula para dejar pasar un líquido, porque solo se le puede cambiar el estado a prendido o apagado. Para lograr este resultado tendrás que programar en lenguaje C la aplicación en el Arduino IDE. Se programará un servidor web el cual alojará la aplicación realizada en HTML que se podrá cargar desde un celular o una computadora, accediendo a esta mediante el IP que tiene el dispositivo NodeMCU ESP8266, ya que estos cuentan con acceso por Wi-Fi.

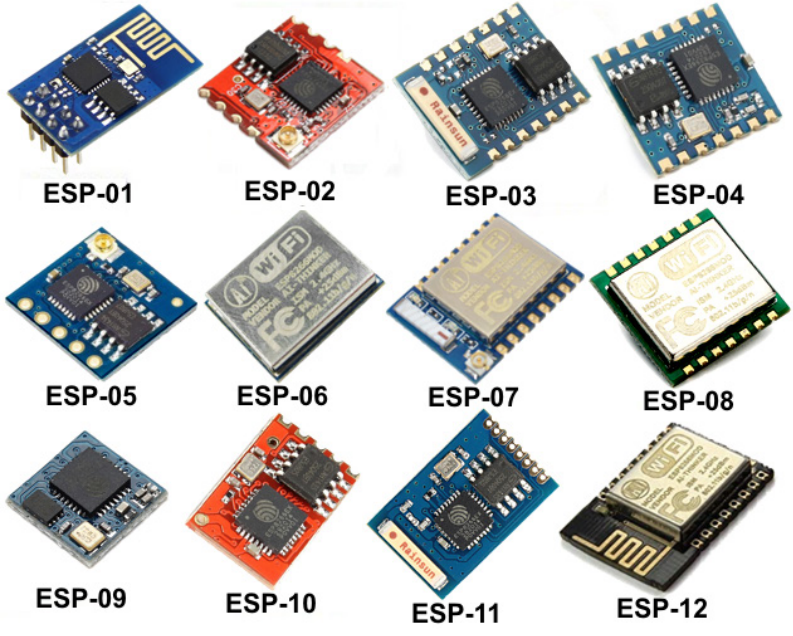
Materiales

1. Un módulo NodeMCU ESP8266.
2. Un cable USB de impresora.
3. Una computadora.
4. Arduino IDE.

ESP8266

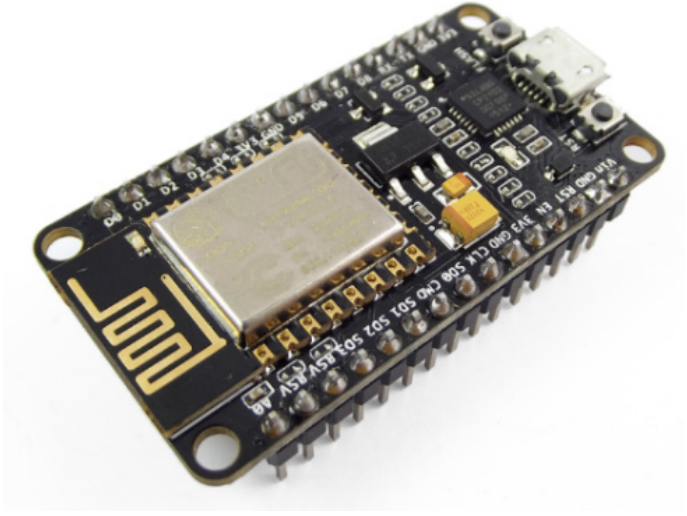
Es un chip que tiene conectividad por Wi-Fi fabricado por la empresa China Espressif, cuenta con un completo *stack* TCP/IP y un microcontrolador. El primero en salir al mercado fue el ESP-01 el cual permite conectarse con una red inalámbrica Wi-Fi mediante el protocolo TCP/IP, utilizando comandos de estilo Hayes conocidos como comandos AT (Valderrama y Brea, 2020). Al igual que el proyecto de Arduino existen muchos modelos que utilizan el ESP8266 como se muestra en las figuras 1 y 2, respectivamente.

Figura 1. Diversos módulos que utilizan el ESP8266



Fuente: Recuperado de http://robots-argentina.com.ar/didactica/wp-content/uploads/ModelosESP.png

Figura 2. NodeMCU ESP8266



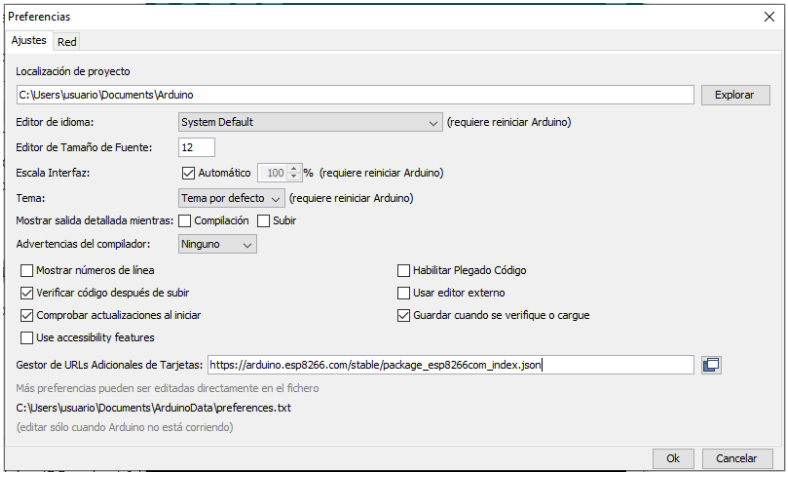
Fuente: Recuperado de http://www.naylampmechatronics.com/474-large_default/nodemcu-esp8266.jpg

Procedimiento

1. Configurar el Arduino IDE para utilizar el módulo NodeMCU ESP8266

Este paso es necesario para que el Arduino IDE conozca cualquiera de las tarjetas que utilizan el ESP8266 en sus diversos modelos; para ello, iremos al menú de Arduino IDE y a la sección de Archivo > Preferencias, en la casilla Gestor de URLs Adicionales de Tarjetas agregamos (*Usando ESP8266 Con El IDE de Arduino*, 2020):

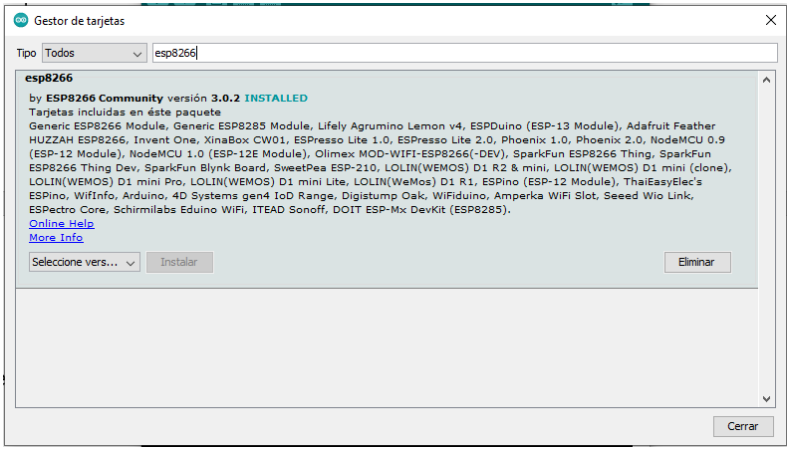
Figura 3. Sección de preferencias de Arduino IDE donde se agregará la URL



Fuente: http://arduino.esp8266.com/stable/package_esp8266com_index.json

Después de agregar la URL ve a la sección Herramientas > placa > Gestor de Tarjetas y busca esp8266 de ESP8266 Community. Procede a instalarla:

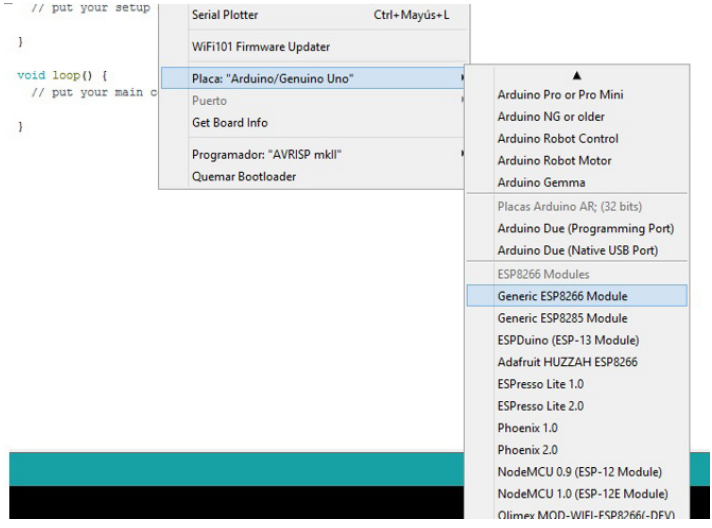
Figura 4. Instalar las tarjetas para placas ESP8266



Fuente: Elaboración propia, captura de pantalla.

La instalación puede tardar un poco, pero una vez que termine puedes revisar en la sección de placa y encontrarás que ya puedes utilizar los módulos del ESP8266, como se muestra en la figura 5.

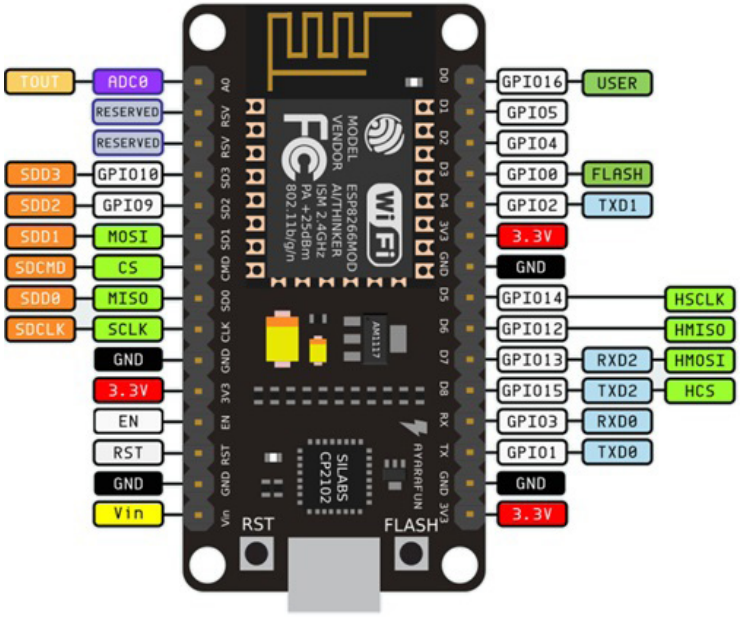
Figura 5. Seleccionar la placa ESP8266
con la que se trabajará



Fuente: Elaboración propia, captura de pantalla.

Selecciona la que vayas a utilizar, esto dependerá del modelo de ESP que tengas. Para esta práctica utilizaremos el NodeMCU ESP8266 versión 0.1 (Parihar, 2019). En la figura 6 se muestra la distribución de los GPIOs (*General Purpose Input/Output*, Entrada/Salida de Propósito General):

Figura 6. Distribución de los pines GPIO del NodeMCU ES8266



Fuente: Recuperado de https://tienda.bricogeek.com/4392-thickbox_default/nodemcu-v3-wifi-esp8266-ch340.jpg

2. Ahora veremos la programación necesaria del NodeMCU8266 para prender y apagar un LED mediante el protocolo http, tecleando el siguiente código al seleccionar la opción de Archivo->Nuevo y luego captura el siguiente programa:

```
//Incluir las librerías para poder utilizar el Wi-Fi
#include <ESP8266WiFi.h>
//Incluir la librería para poder crear nuestro servidor Web
#include <ESP8266WebServer.h>

//Hay que darle un nombre y contraseña a nuestro módulo en la red Wi-Fi
const char *NOMBRE_RED = “Practica2”,
```

```
        *CLAVE_RED = "12345678";
//Definimos nuestro puerto donde arrancará el servidor Web
//y usaremos el led que viene inter construido en el NodeMCU
const int PUERTO = 80,
        LED = LED_BUILTIN;

// Tener un estado del LED
bool estadoDelLED = false;

//Ahora configuraremos la IP, puerta de enlace y máscara de red
de nuestro dispositivo
IPAddress ip(192,168,4,22);
IPAddress puertaDeEnlace(192, 168, 4, 9);
IPAddress mascaraDeRed(255, 255, 255, 0);

//Arrancamos el servidor web dentro de nuestro dispositivo
ESP8266WebServer servidor(PUERTO);

//Esta función crea la interfaz en HTML
//Que se mostrará al conectarnos a la IP del dispositivo
String obtenerInterfaz()
{
   String HtmlRespuesta = "<!DOCTYPE html>\n<html>\
n<head>\n \"<meta name='viewport' content='width=de-
vice-width, initial-scale=1.0'>\"\n<style>\n.buttonprender
{\n background-color: #4CAF50;\n border: none;\n color:
white;\n padding: 15px 32px;\n text-align: center;\n text-dec-
oration: none;\n display: inline-block;\n font-size: 16px-
;\n margin: 4px 2px;\n cursor: pointer;\n}\n\n.buttonapagar
{\n background-color: #CD5C5C;\n border: none;\n color:
white;\n padding: 15px 32px;\n text-align: center;\n text-deco-
ration: none;\n display: inline-block;\n font-size: 16px;\n mar-
gin: 4px 2px;\n cursor: pointer;\n}\n\n</style>\n</head>\
n<body>";

 HtmlRespuesta += "<h1>Encender y apagar LED</h1>";
 HtmlRespuesta += "<p>El LED est&aacute; actualmente ";
```

```
 if (estadoDelLED)
 {
  HtmlRespuesta += "encendido</p>";
                HtmlRespuesta += "<a href='/apagar'
class='buttonapagar'>Apagar</a>";
 }
 else
 {
  HtmlRespuesta += "apagado</p>";
               HtmlRespuesta += "<a href='/encender'
class='buttonprender'>Encender</a>";
 }
 HtmlRespuesta += "<br><a href='https://telematicanet.ucol.
mx'>Universidad de Colima</a>";
 HtmlRespuesta += "</body>"
           "</html>";
 return HtmlRespuesta;
}

//cuando los usuarios se conecten mostrará la Interfaz HTML
void mostrarInterfazHTML()
{
 servidor.send(200, "text/html", obtenerInterfaz());
}

// Si selecciona la ruta de encendido prenderá el LED
void rutaEncenderLed()
{
 estadoDelLED = true;
 digitalWrite(LED, LOW); //porque el LED interno se comporta de
manera invertida
 mostrarInterfazHTML();
}


//Si selecciona la ruta de apagado, apagará el LED
void rutaApagarLed()
```

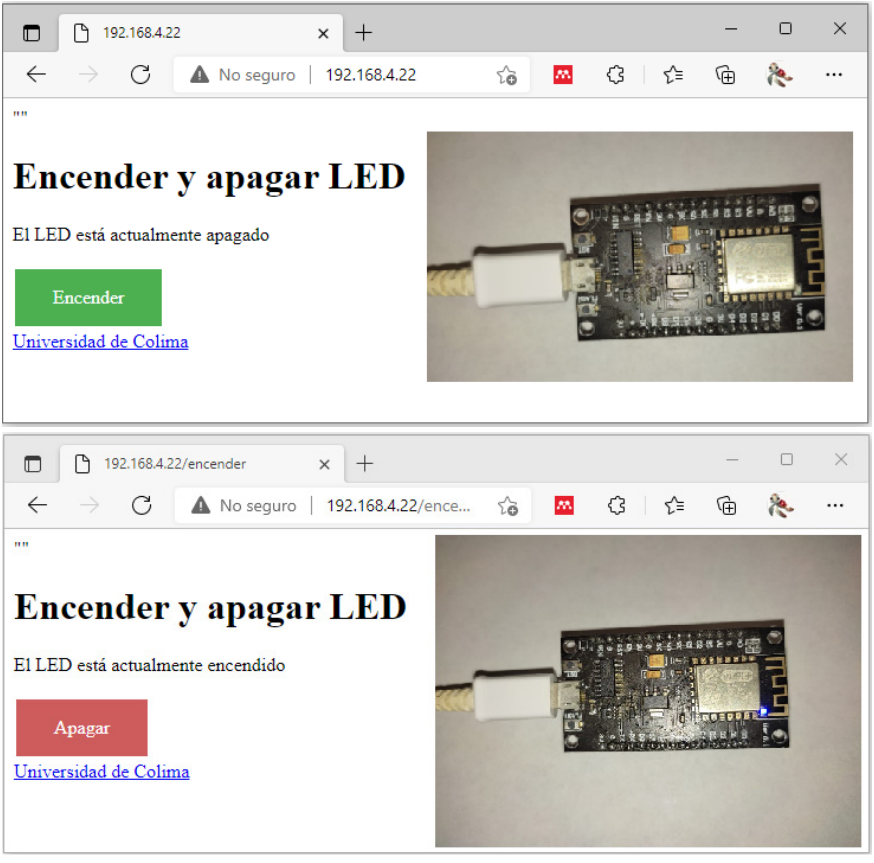
```
{
 estadoDelLED = false;
 digitalWrite(LED, HIGH);//por que el LED interno se comporta de manera invertida
 mostrarInterfazHTML();
}

//Y si selecciona la ruta raíz muestra la interfaz para apagar y encender el LED
void rutaRaiz()
{
 mostrarInterfazHTML();
}

//En caso de que de una ruta que no existe, le mostrará error de que no existe el recurso
void rutaNoEncontrada()
{
 servidor.send(404, "text/plain", "404");
}

void setup()
{
 Serial.begin(115200);
 Serial.println();
// Configuración del LED para salida
 pinMode(LED, OUTPUT);

 //Se establecerá la configuración como punto de acceso
 Serial.print("Establecer la configuración de Soft-AP ... ");
 Serial.println(WiFi.softAPConfig(ip, puertaDeEnlace, mascaraDeRed) ? "Ready" : "Failed!");

 Serial.print("Se configurará el nombre y la contraseña de la red del Soft-AP ... ");
     Serial.println(WiFi.softAP(NOMBRE_RED, CLAVE_RED) ? "Ready" : "Failed!");
```

```
  Serial.print("La dirección del Soft-AP = ");
  Serial.println(WiFi.softAPIP());
  delay(100);

  // Configuramos la ruta y la función que responderá a la solicitud
de dicha ruta
  servidor.on("/", rutaRaiz);
  servidor.on("/encender", rutaEncenderLed);
  servidor.on("/apagar", rutaApagarLed);
  servidor.onNotFound(rutaNoEncontrada);

  // Empezar a escuchar a los usuarios que se quieren conectar por
el protocolo http
  servidor.begin();
}

// En el loop manejamos a los clientes conectados
void loop()
{
  servidor.handleClient();
}
```

3. Para guardar el programa en la computadora debes de seleccionar del menú Archivo->Salvar, asígnale el nombre al *sketch* o presionando las teclas Ctrl más la tecla S (Ctrl+S).
4. Ahora debes compilar el programa seleccionando del menú Programa->Verificar/Compilar o presionando las teclas Ctrl más la tecla R (Ctrl+R), o del IDE el botón que tiene una flecha .
5. Si no hubo ningún problema al compilarlo, ya puedes subir tu programa a la placa Arduino, y ver el resultado, para ello seleccionar del menú Programa->Subir, otra opción es presionar las teclas Ctrl más la tecla U (Ctrl+U), o del IDE el botón que tiene una flecha hacia la derecha , después de ello podrás ver cómo enciende

y se apaga el LED que conectaste en tu placa NodeMCU ESP8266 por la IP, seleccionando la red y luego abriendo el navegador, prosiguiendo a teclear la IP, como puedes verificar en la figura 7.

Figura 7. Capturas de pantalla e imágenes de la aplicación web para encender y apagar el LED



Fuente: Elaboración propia, captura de pantallas.

Ya solo resta que realices todas las actividades del procedimiento y que pruebes tu práctica.

Práctica: prender y apagar un led con NodeMCU ESP8266 utilizando el protocolo MQTT

Objetivo general

Encender y apagar un LED utilizando el módulo NodeMCU ESP8266 mediante el protocolo MQTT. En esta práctica trabajarás la electrónica que permite encender o apagar un LED, en este caso será el que está interconstruido en la propia placa para fines prácticos. Podemos considerar que los LEDs son como los actuadores o accionadores, pues con ellos podríamos prender una luminaria, una bomba o abrir una electroválvula para dejar pasar un líquido, porque solo se le puede cambiar el estado de prendido o apagado. Para lograr este resultado tendrás que programar en lenguaje C la aplicación en el Arduino IDE, se programará la placa para que esta se conecte por Wi-Fi a Internet, lo que le permitirá conectarse a un *broker* MQTT (programa servidor que recibe y renvía mensajes a todos los subscriptores que estén conectados a un tema específico), esto lo realiza a través de *websockets* (tecnología que proporciona una comunicación *full*-dúplex), lo que hace que la comunicación se sienta como en tiempo real, y permitirá enviar el mensaje de prender o apagar desde la aplicación web al *broker* MQTT. Este a su vez enviará el mensaje al dispositivo NodeMCU ESP8266 que al recibirlo prenderá el LED de su placa o lo apagará (Mishra y Kertesz, 2020).

Materiales

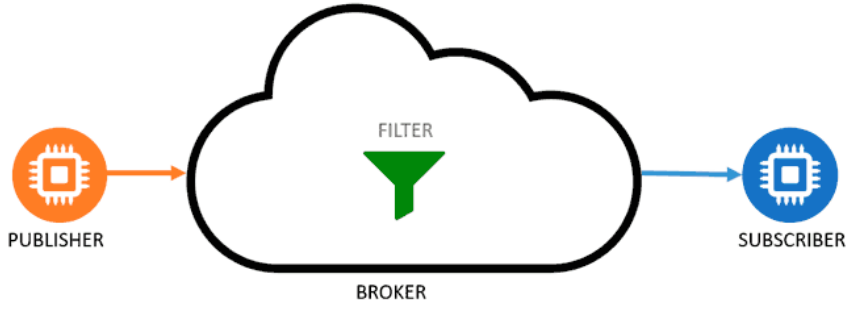
1. Un módulo NodeMCU ESP8266.
2. Un cable USB de impresora.
3. Una computadora.
4. Arduino IDE.

Protocolo MQTT

En esta práctica utilizaremos el protocolo MQTT es un protocolo de comunicación *machine to machine,* de maquina a máquina (M2M) que tiene una gran popularidad para la comunicación entre dispositivos IoT.

Este protocolo es uno de los principales del IoT debido a su sencillez y ligereza, lo cual es muy importante ya que estos dispositivos tienen limitantes como lo son de consumo, potencia y ancho de banda (ver figura 1). El MQTT (*Message Queing Telemetry Transport*), se basa en la pila o *stack* TCP/IP. Como su base para la comunicación es diferente a una comunicación HTTP donde cada petición se realiza mediante una conexión, mientras que MQTT es un servicio de mensajería de tipo *push* donde es necesaria la subscripción y la publicación, la que se realiza conectándose a un servidor centrar llamado *broker*. La clasificación de estos mensajes se realiza mediante *topics* (temas), que son organizados de manera jerárquica, por lo que los clientes pueden publicar mensajes al tema de su interés, y otros clientes pueden subscribirse a esos temas, de manera que el *broker* les hará llegar los mensajes.

Figura 1. Funcionamiento del protocolo MQTT



Fuente: Recuperado de https://www.luisllamas.es/wp-content/uploads/2019/02/protocolos-iot-pubsub.png

Esto significa que los clientes se mantienen comunicados con el *broker* por el protocolo TCP/IP el cual mantiene un registro de los subscriptores, la comunicación es mantenida abierta hasta que los clientes se desconectan de ella, el puerto que utiliza esta comunicación es 1883 y a veces 8883 que es de tipo TSL (del inglés *Transport Layer Security*, protocolo de seguridad de la capa de transporte), pero se puede configurar cualquier puerto en el *broker*.

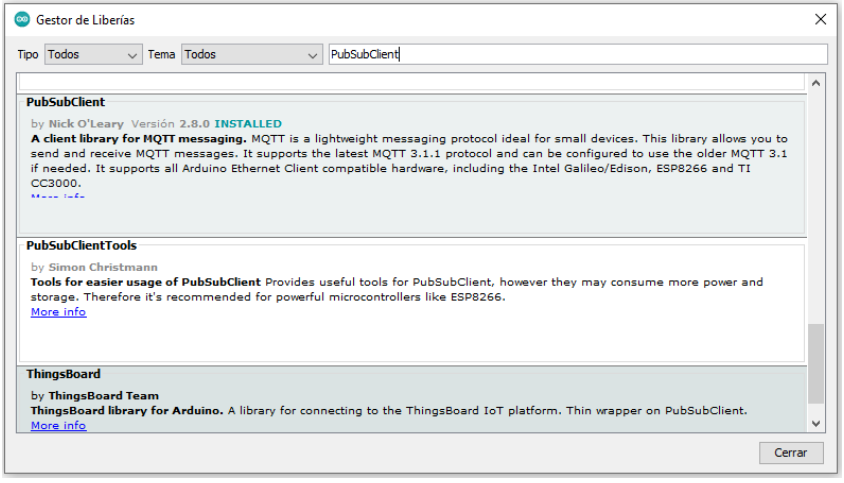
Eclipse Mosquitto MQTT *broker*

Será el servidor de mensajes de código abierto (*open source*) que utilizaremos en esta práctica el cual tiene implementadas las versiones del protocolo MQTT 3.1 y 3.1.1 protocolos ligeros lo que permitirán conectar nuestro dispositivo con celulares o computadoras, no instalaremos nuestro propio servidor, solo utilizaremos el que la organización provee para testear nuestras aplicaciones que se encuentra en la dirección test.mosquitto.org en el puerto 1883 (Hwang et al., 2019).

Procedimiento

1. Debes configurar tu Arduino IDE para que pueda trabajar con el módulo o tarjeta NodeMCU ESP8266.
2. Agrega la librería para hacer uso del protocolo MQTT llamada *PubSubClient* (Almenara, 2020), para ello ve a la sección del menú del Arduino IDE Programa->Incluir Librería->Administración de Biblioteca… y busca la librería tecleando su nombre PubSubClient, luego instálala, como se muestra en la imagen de la figura 2.

Figura 2. Captura de pantalla de como agregar la librería PubSubClient desde Arduino IDE



Fuente: Elaboración propia, captura de pantalla.

3. Ahora veremos la programación necesaria del NodeMCU8266 para prender y apagar un LED mediante el protocolo http, tecleando el siguiente código y seleccionando la opción de Archivo->Nuevo y luego captura el siguiente programa:

```
//Librería para podernos conectar el dispositivo a Internet
//por Wi-Fi
#include <ESP8266WiFi.h>
//Librería para podernos conectar al broker MQTT
#include <PubSubClient.h>

//Constantes para conectarnos a la red Wi-Fi mediante ssid y su contraseña
const char* ssid = "AP-PTCs";
const char* password = "tc-ptcs2014";
```

```
//Dirección del broker MQTT
const char* mqtt_server = "test.mosquitto.org";

//Aquí se configura la conexión del dispositivo para conectarse
//al Wi-Fi y al broker
WiFiClient espClient;
PubSubClient client(espClient);
unsigned long lastMsg = 0;
#define MSG_BUFFER_SIZE (50)
char msg[MSG_BUFFER_SIZE];
int value = 0;

void setup_wifi() {

  delay(10);
  // Empezamos por conectarnos a una red WiFi
  Serial.println();
  Serial.print("Conectado a ");
  Serial.println(ssid);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  randomSeed(micros());

  Serial.println("");
  Serial.println("WiFi conectado");
  Serial.println("Dirección IP: ");
  Serial.println(WiFi.localIP());
```

```
}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Llegó el mensaje del tema [");
  Serial.print(topic);
  Serial.print("] ");

  String message;
  for (int i = 0; i < length; i++) {
    message = message + (char) payload[i];  // convert *byte to string
    Serial.print((char)payload[i]);
  }
  Serial.println();
  Serial.print(message);

  if(message == "prender")
  {
  digitalWrite(BUILTIN_LED, LOW);  // Ponemos el voltaje del LED en
bajo porque el dispositivo lo tiene invertido
  Serial.println();
  Serial.println("Se prendio el LED");
  }

  if(message == "apagar")
  {
  digitalWrite(BUILTIN_LED, HIGH);   // Ponemos el voltaje del LED
en alto porque el dispositivo lo tiene invertido
  Serial.println("Se apago el LED");
  }
}

void reconnect() {
  // Bucle hasta que nos volvamos a conectar
  while (!client.connected()) {
```

```
    Serial.print("Intentando la conexión MQTT...");
    // Crea una identificación de cliente aleatoria
    String clientId = "ESP8266Client-";
    clientId += String(random(0xffff), HEX);
    // Intenta conectarse
    if (client.connect(clientId.c_str())) {
      Serial.println("conectado");
      // Una vez conectado, publique un anuncio...
      client.publish("arg/led", "hello world");
      // ... y volver a suscribirte
      client.subscribe("arg/acciones");
    } else {
      Serial.print("fallido, rc =");
      Serial.print(client.state());
      Serial.println(" inténtalo de nuevo en 5 segundos");
      // Espere 5 segundos antes de volver a intentarlo
      delay(5000);
    }
  }
}

void setup() {
  pinMode(BUILTIN_LED, OUTPUT);     // Inicialice el pin BUILTIN_
LED como salida
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}

void loop() {

  if (!client.connected()) {
    reconnect();
```

```
  }
  client.loop();

}
```

4. Para guardar el programa en la computadora selecciona en el menú Archivo->Salvar y asígnale el nombre al *sketch,* o bien presiona las teclas Ctrl más la tecla S (Ctrl+S).
5. Ahora debes compilar el programa seleccionando del menú Programa->Verificar/Compilar o presionando las teclas Ctrl más la tecla R (Ctrl+R), otra opción es del IDE seleccionar el botón que tiene una flecha .
6. Si no hubo ningún problema al compilarlo, ya puedes subir tu programa a la placa Arduino, y ver el resultado, para ello selecciona del menú Programa->Subir o presiona las teclas Ctrl más la tecla U (Ctrl+U), otra opción es del IDE seleccionar el botón que tiene una flecha hacia la derecha . Después de ello podrás ver cómo enciende y se apaga el LED que conectaste en tu placa NodeMCU ESP8266 utilizando la aplicación web que tienes que programar y que se describe en el siguiente punto.
7. Ahora tendrás que crear la aplicación web que utilice una librería para conectarse a un servidor MQTT, de manera que pueda suscribirse a un canal donde se enviará el mensaje de prender y apagar, que será recibido por el NodeMCU el cual prenderá o apagará el LED según el mensaje que reciba desde la aplicación. Enseguida se muestra el código:

```
<!DOCTYPE html>
<html>
  <head>
   <!-- Utilizar la librería para poder utilizar el protocolo MQTT-->
   <script src="https://cdnjs.cloudflare.com/ajax/libs/mqtt/4.2.6/
```

```
mqtt.js" > < /script >
< script >

   //Conectarse al broker mosquito
   var client = mqtt.connect("ws://test.mosquitto.org:8080/mqtt");


//subscribirse al canal arg, mi canal
function EventoConectar() {
 console.log("Conectado a MQTT");
 client.subscribe("arg/#", function (err) {
  if (!err) {
    //enviar el mensaje al broker a mi topic arg subtema acciones el
mensaje de apagar
    //para que inicie con el LED apagado
    client.publish("arg/acciones", "apagar");
  }
 });
}

//función que recibe los mensajes del broker que son del topic de mi
interés
function EventoMensaje(topic, message) {
 if (topic = = "arg/acciones") {
  console.log("Acción " + message.toString());
 }
 console.log(topic + " - " + message.toString());
 // client.end()
}

//Conectarse al broker
client.on("connect", EventoConectar);
//Ponerse a la escucha de la llegada de mensajes del broker
client.on("message", EventoMensaje);
< /script >
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
<!--Definir estilos de mi aplicación para mostrar un botón de switch-->
<style>
.switch {
  position: relative;
  display: inline-block;
  width: 60px;
  height: 34px;
}

.switch input {
  opacity: 0;
  width: 0;
  height: 0;
}

.slider {
  position: absolute;
  cursor: pointer;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  background-color: #ccc;
  -webkit-transition: .4s;
  transition: .4s;
}

.slider:before {
  position: absolute;
  content: "";
  height: 26px;
  width: 26px;
```

```
  left: 4px;
  bottom: 4px;
  background-color: white;
  -webkit-transition: .4s;
  transition: .4s;
}

input:checked + .slider {
  background-color: #2196F3;
}

input:focus + .slider {
  box-shadow: 0 0 1px #2196F3;
}

input:checked + .slider:before {
  -webkit-transform: translateX(26px);
  -ms-transform: translateX(26px);
  transform: translateX(26px);
}

/* Rounded sliders */
.slider.round {
  border-radius: 34px;
}

.slider.round:before {
  border-radius: 50%;
}
</style>
</head>
<body>
<script>
```

```
//iniciar el estado del botón en apagado
prendido = false;
//función que cambia de prendido a apagado o viceversa
function mifuncion()
{
   prendido = !prendido;
if(prendido){
//enviar el mensaje al broker para prender el LEd
client.publish("arg/acciones", "prender");
document.getElementById("estado").innerHTML = "Encendido";
}
else
{
//Enviar el mensaje al broker para apagar el LED
client.publish("arg/acciones", "apagar");
document.getElementById("estado").innerHTML = "Apagado";
}
}

</script>
<!--Mostrar la interfaz principal de mi aplicación web-->
   <h2>Prender y apagar un LED con el protocolo MQTT</h2>
   <h2>y la placa NodeMCU 8266</h2>
   <label class = "switch">
     <input type = "checkbox" onclick = "mifuncion()">
     <span class = "slider"></span>
   </label>
   <h1 id = "estado"></h1>
</body>
</html>
```

8. Para este caso, a la aplicación se le dio el nombre de clientemqtt.html, edítalo y guárdalo, para ello te sugiero utilizar el editor de código Visual Studio Code, puedes

usar Atom, SublimeText entre otros. Para ejecutarlo, agregamos entonces la extensión iServe que permite ejecutar en un servidor que trae como *plugin* este editor. Es suficiente con abrir en el navegador el archivo y así podrás ejecutar en tu navegador la aplicación que se verá como en la figura 3.

Figura 3. Capturas de pantalla de la aplicación web para encender y apagar un LED

Fuente: Elaboración propia, capturas de pantalla.

Se puede observar que aparece un botón de *switch* (de prender y apagar) con el cual puedes encender y apagar el led interconstruido

que trae el nodeMCU ESP8266, ya nada más queda que pruebes esta práctica y veas tú mismo los resultados. Un buen reto importante es que montes tu propio servidor de mosquitto (*broker* mqtt).

Práctica: enviar y recibir datos desde la plataforma de IoT Ubidots

Objetivo general

El objetivo de esta práctica es el envío y recepción de datos utilizando la plataforma IoT Ubidots, para ello se conectará un sensor análogo LM35 de temperatura al NodeMCU ESP8266, la cual será enviada hacia la plataforma y desde UBIDOTS se colocará un *widget* tipo botón para prender y apagar que enviará un 1 o un 0 respectivamente, para controlar el LED interconstruido en el módulo NodeMCU. Para ello hay que conectar este por medio de Wi-Fi a Internet, a fin de realizar las acciones señaladas.

Materiales

1. Un módulo NodeMCU ESP8266.
2. Un cable USB de impresora.
3. Una computadora.
4. Sensor de temperatura LM35.
5. Cables dupont.
6. Arduino IDE.

Ubidots

Es un servicio que está en la nube y te permite tareas como almacenar valores de sensores, interactuar mediante tableros donde se pueden poner *widgets* (componentes) realizar acciones que pueden cambiar valores de variables a fin de enviarse a los dispositivos para tomar decisiones en ellos. Los datos pueden enviarse por el Protocolo TCP/IP directamente o usar su servidor MQTT. Esto lo puedes realizar directamente desde su página web o app, incluso puedes exportar sus componentes en formato web para ser embebidos en tu propio sistema o aplicación móvil, como se aprecia en la figura 1 (Álvarez Carulla, 2021).

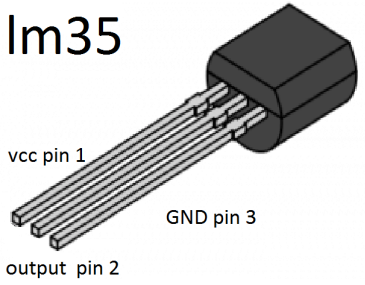
Figura 1 . Plataforma Ubidots para la implementación d e proyectos de IoT

Fuente: Elaboración propia, captura de pantalla.

LM35

Sensor de bajo precio, además de no requerir otro circuito para lograr su funcionamiento; algunos de sus usos son el simple monitoreo de la temperatura, usarlo en termostatos y como termómetro. Su salida es analógica y maneja rangos de temperatura de 55 hasta 150 grados centígrados el cual está calibrado de fábrica, este cuenta con una exactitud de 0.5ºC, hay que alimentarlo con 5V (Junizan et al., 2019), cuya imagen está en la figura 2.

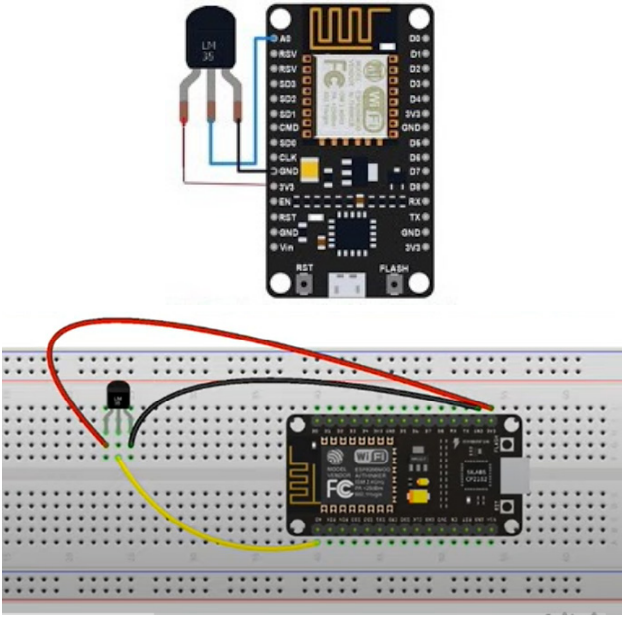
Figura 2. Sensor de temperatura LM35



Fuente: Recuperado de https://hetpro-store.com/TUTORIALES/wp-content/uploads/2016/05/lm35-img.png

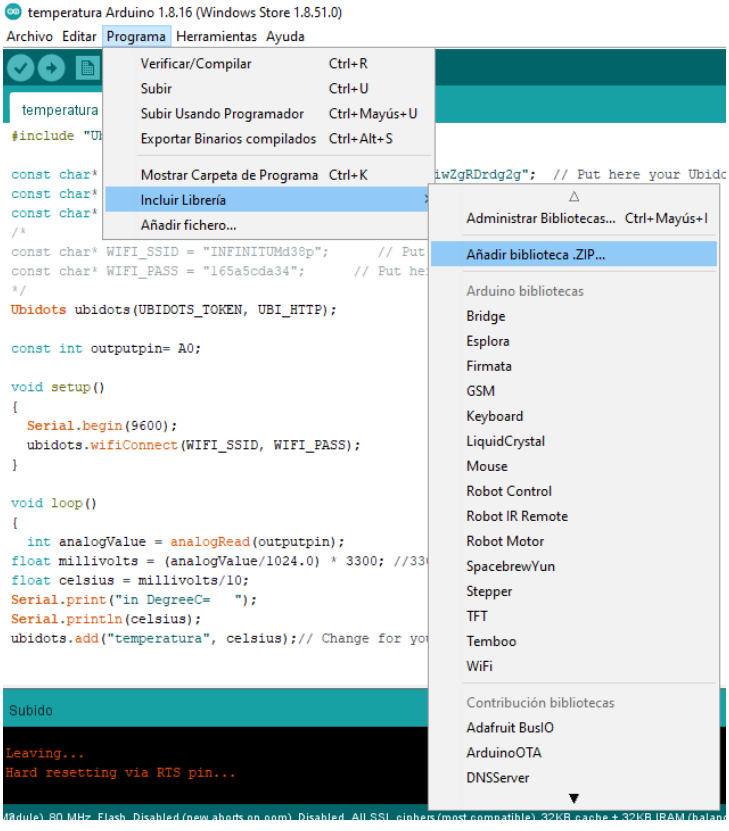
A continuación, en la figura 2, se muestra cómo interconectar el sensor de temperatura LM35, para que lo puedas utilizar en esta práctica.

Figura 2. Esquema de cómo interconectar el LM35 al NodeMCU ESP8266



Fuente: Recuperado de https://www.youtube.com/watch?v=-oLfthiXieU

Figura 4. Captura de pantalla de opción para agregar la librería de Ubidots



Fuente: Elaboración propia, captura de pantalla.

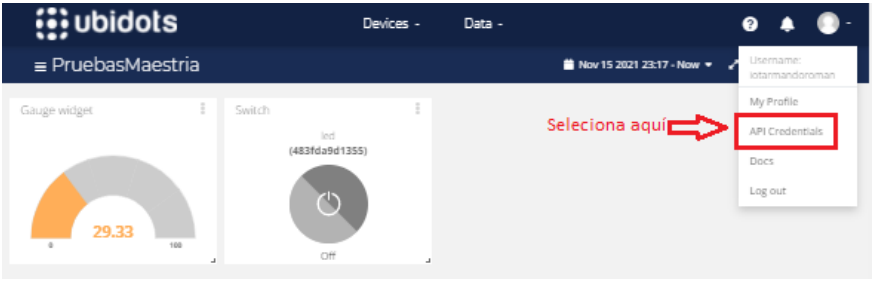
Procedimiento

1. Debes configurar tu Arduino IDE para que pueda trabajar con el módulo o tarjeta nodeMCU ESP8266.
2. Agrega la librería para hacer uso del protocolo de la plataforma Ubidots, llamada *ubidots-esp8266-master.zip* que se descarga de la dirección https://github.com/ubidots/ubidots-nodemcu/archive/master.zip, después dirígete a la sección del menú del Arduino IDE Programa->Incluir Li-

brería- > Añadir biblioteca.Zip… y busca la librería que descargaste, selecciónala para instalarla (ver figura 4).

3. Para utilizar la plataforma de Ubidots tendrás que registrarte, en ella debes identificar dónde se encuentra el *token* que te permitirá conectar tu dispositivo con la plataforma. En la figura 5 se muestra dónde se encuentra en Ubidots.

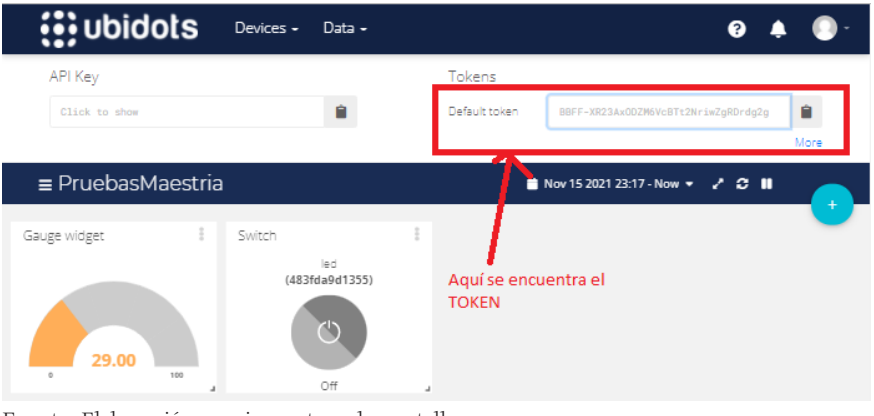
Figura 5. Captura de pantalla de opción para seleccionar API Credential



Fuente: Elaboración propia, captura de pantalla.

Luego seleccionamos el token, cópialo ya que lo necesitarás para agregarlo al programa de tu dispositivo, a fin de enviar y recibir información a Ubidots. En la Figura 6 se observa dónde podrás encontrar este elemento, resaltándolo en un marco rojo:

Figura 6. Ubicación del *token*



Fuente: Elaboración propia, captura de pantalla.

4. Ahora veremos la programación necesaria del NodeMCU8266 para enviar los datos de la temperatura hacia la plataforma IoT de Ubidots y recibir el valor del *widget,* a fin de prender y apagar el LED interconstruido, tecleando el siguiente código y seleccionando la opción de Archivo->Nuevo, luego captura el siguiente programa:

```
#include "Ubidots.h"

//Indicar el token de nuestra plataforma Ubidots para conectarnos a
nuestra cuenta
const char* UBIDOTS_TOKEN = "BBFF-XR23AxODZM6VcBTt2N-
riwZgRDrdg2g";  // Poner aqui el TOKEN

//definir el nombre de nuestro punto de acceso y su contraseña para
salir a Internet
const char* WIFI_SSID = "AP-PTCs";      // Put here your Wi-Fi SSID
const char* WIFI_PASS = "tc-ptcs2014";       // Put here your Wi-Fi
password

//Inicializar a qué cuenta nos conectaremos a Ubidots
//y con qué protocolo
Ubidots ubidots(UBIDOTS_TOKEN, UBI_HTTP);

//Seleccionar el pin donde se conectará el sensor de temperatura
//lm35
const int outputpin= A0;

void setup()
{
  pinMode(BUILTIN_LED, OUTPUT);    // Inicialice el pin BUILTIN_
LED como salida
  //Conectarse a Internet
  Serial.begin(9600);
```

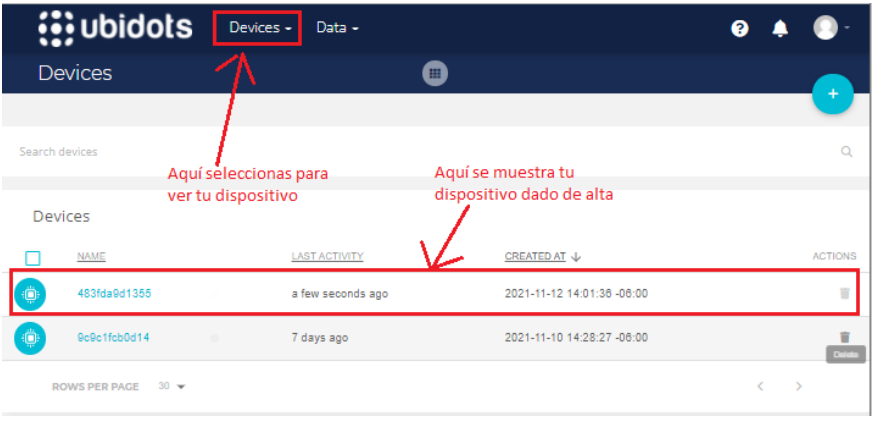
```
  ubidots.wifiConnect(WIFI_SSID, WIFI_PASS);
}

void loop()
{
//Calcular la temperatura
//obteniendo el valor analógico de la entrada A0
//donde se encuentra conectado el sensor LM35
int analogValue = analogRead(outputpin);
float millivolts = (analogValue/1024.0) * 3300; //3300 is the voltage
provided by NodeMCU
float celsius = millivolts/10;
Serial.print("in DegreeC=   ");
Serial.println(celsius);
ubidots.add("temperatura", celsius);// Preparar la variable tempera-
tura y su valor a enviar

  bool bufferSent = false;
  bufferSent = ubidots.send(); // Enviar variable a Ubidots
  if (bufferSent) {
  // Se enviaron los valores correctamente
   Serial.println("Values sent by the device");
  }

  //leer estado del botón de Ubidots
  /* Obtain last value from a variable as float using HTTP */
  float value = ubidots.get("483fda9d1355", "led");

  // Evaluates the results obtained
  if (value != ERROR_VALUE) {
    Serial.print("Valor del boton:");
    Serial.println(value);
  }
```

```
 if(value = = 1)
 {
  digitalWrite(BUILTIN_LED, LOW);  // Ponemos el voltaje del LED
en bajo porque el dispositivo lo tiene invertido
 Serial.println();
 Serial.println("Se prendio el LED");
 }

 if(value = = 0)
 {
 digitalWrite(BUILTIN_LED, HIGH);  // Ponemos el voltaje del LE-
Den alto porque el dispositivo lo tiene invertido
 Serial.println("Se apago el LED");
 }
 delay(5000);
}
```

5. Para guardar el programa en la computadora selecciona del menú Archivo->Salvar y asígnale el nombre al *sketch*, o presiona las teclas Ctrl más la tecla S (Ctrl + S).
6. Ahora debes compilar el programa seleccionando del menú Programa->Verificar/Compilar o presionando las teclas Ctrl más la tecla R (Ctrl + R), otra opción es desde el IDE seleccionar el botón que tiene una flecha .
7. Si no hubo ningún problema al compilarlo, ya puedes subir tu programa a la placa Arduino y ver el resultado, para ello selecciona del menú Programa->Subir o presiona las teclas Ctrl más la tecla U (Ctrl + U), otra opción es desde el IDE seleccionar el botón que tiene una flecha hacia la derecha , después de ello podrás ver cómo se enciende y se apaga el LED interconstruido del NodeMCU ESP8266 cuando utilices el *dashboard*, donde se encuentra el *widget switch* para realizar esta acción; además, podrás ver también en el *widget gauge* cómo se afecta por el envío de datos del dispositivo con la temperatura censada del LM35.

8. Después de esto verás reflejado tu dispositivo NodeMCU ESP8266 en Ubidots. En la figura 7 se muestra dónde observarás tu dispositivo seleccionando del menú la sección de *Devices*:

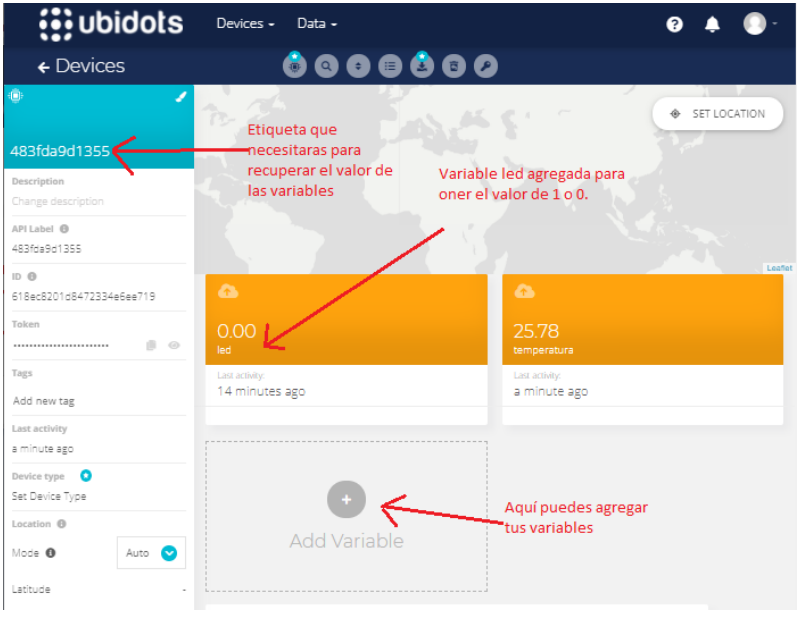
Figura 7. Ubicación de Devices



Fuente: Elaboración propia, captura de pantalla.

9. Ahora es momento de agregar el *widget* que te permitirá prender y apagar el LED Interconstruido de tu NodeMCU ESP8266, para ello, en la sección de *Devices* seleccionas tu dispositivo, en esta sección podrás añadir nuevas variables, en este caso será “LED”, como se muestra en la figura 8.

Figura 8. Captura de pantalla que muestra cómo agregar las variables y el nombre de dispositivo

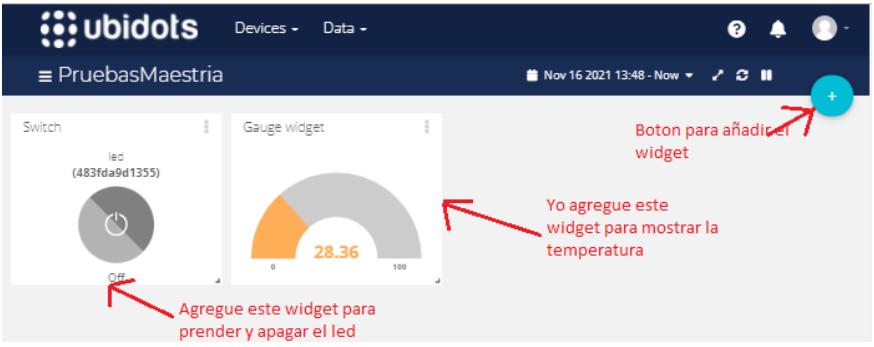


Fuente: Elaboración propia, captura de pantalla.

En la figura 8 se muestra también lo que se conoce como el *label* o etiqueta que tiene el nombre del dispositivo, esta te servirá para recuperar los valores de las variables; en este caso la llamada "LED" que tendrá un 0 o un 1 el cual será asignado mediante un *widget* del *dashboard* o panel de control de Ubidots.

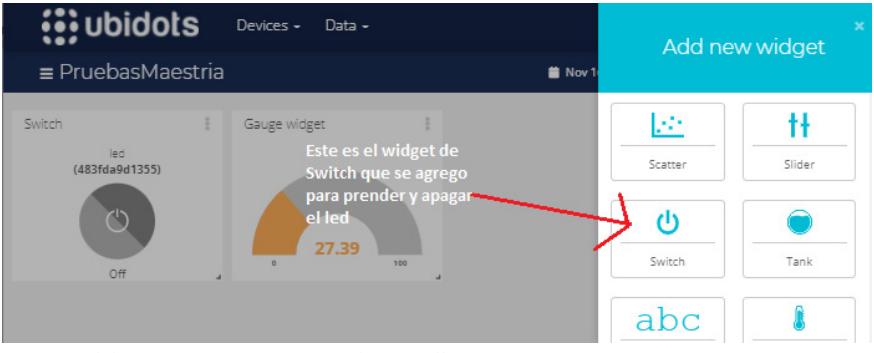
10. Para agregar el widget tendrás que ir a la sección Data->Dashboards del menú de Ubidots, ahí con el botón que tiene el símbolo + agrega el *widget* de prender y apagar, para ello, te pedirá agregar la variable que afectará el widget, en este caso será la variable "LED", en las figuras 9 y 10 se muestra este proceso.

Figura 9. Ubicación de botón para añadir widget



Fuente: Elaboración propia, captura de pantalla.

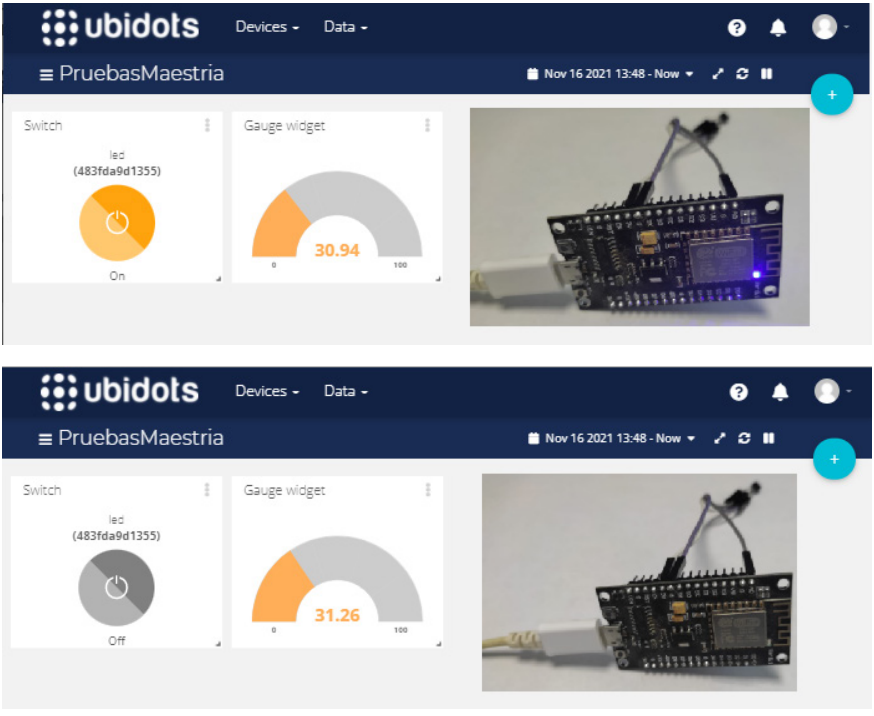
Figura 10. Ubicación del menú de widgets



Fuente: Elaboración propia, captura de pantalla.

11. Una vez agregados los *widgets* puedes ver cómo el de temperatura cambia si hay variaciones de temperatura, así como si prendes o apagas con el *widget switch* ello tendrá un efecto en el LED inter construido del NodeMCU ESP8266 para realizar estas acciones, como se muestra en la figura 11.

Figura 11. Pantalla que muestra el efecto
que tienen los *widgets* para prender y apagar el LED



Fuente: Elaboración propia, captura de pantalla.

Práctica: enviar datos a una base de datos de MySQL desde NodeMCU

Objetivo general

El objetivo de esta práctica es el envío de datos del sensor BMP180 conectado en un NodeMCU hacia un servidor de base de datos MySql ubicado en Internet. Esto se realiza conectando un sensor digital BMP18, el cual sensará las variables de temperatura, altitud y presión barométrica; entonces el NodeMCU ESP8266, por medio del protocolo http utilizando el método GET, enviará estos valores a la base de datos hospedada en un sitio que tiene una aplicación reali-

zada en PHP, la cual recibe los datos y los almacena en una base de MySql, para ello hay que conectarlo por medio de Wi-Fi a Internet a fin de realizar las acciones señaladas (Abdullah et al., 2021).

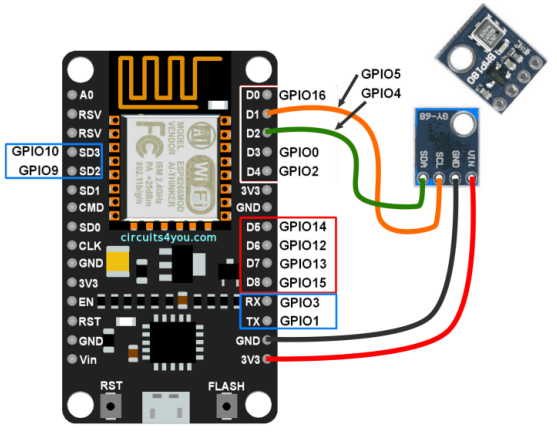
Materiales

1. Un módulo NodeMCU ESP8266.
2. Un cable USB de impresora.
3. Una computadora.
4. Sensor de temperatura BMP180.
5. Cables dupont.
6. Hospedaje Web que cuente con el servidor Apache y MySQL.
7. Arduino IDE.

Sensor BMP180

Sensor de presión absoluta llamada también barométrica, fue diseñado para leer la presión atmosférica y, con base en este valor, puede calcular de manera indirecta la altitud sobre el nivel del mar, la presión atmosférica varía con respecto a la temperatura y la altitud, por lo que este sensor mide también la temperatura de tal forma que puede compensar la influencia que esta tiene sobre la presión y así permite determinar con mayor presión la altitud. En la siguiente figura se muestra cómo es el sensor bmp180 y su interconexión con el módulo NodeMCU ESP82266 (Ilham, 2020) que se observa en la figura 1.

Figura 1. Sensor BMP180 y su interconexión con el módulo NodeMCU ESP8266



Fuente: Recuperado de https://circuits4you.com/wp-content/uploads/2019/03/NodeMCU-ESP8266-connections-with-BMP180.png

Procedimiento

1. Primero tenemos que crear la base de datos en MySQL para almacenar los datos de temperatura, presión, altitud, fecha y hora de registro de la lectura, para ello se utilizó en este caso un servidor de hospedaje de base de datos y páginas web gratuito creado en https://mex.000webhost.com/, ahí creamos nuestra base de datos y tabla que tiene la estructura que se muestra en la figura 2.

Figura 2. Estructura de la tabla

	#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
☐	1	id	int(11)			No	*Ninguna*		AUTO_INCREMENT	Cambiar Eliminar Más
☐	2	temperatura	float			No	*Ninguna*			Cambiar Eliminar Más
☐	3	presion	float			No	*Ninguna*			Cambiar Eliminar Más
☐	4	altitud	float			No	*Ninguna*			Cambiar Eliminar Más
☐	5	fecha	datetime			No	*Ninguna*			Cambiar Eliminar Más

Fuente: Elaboración propia, captura de pantalla.

2. Ahora tendremos que crear un API (del inglés *application programming interfaz*, programa que nos permite acceder desde Internet a la base de datos para insertar los registros) en PHP (lenguaje de programación de propósito general para desarrollo web). A continuación, mostramos el código fuente correspondiente:

```
<?php
//Selecionar la zona horaria de México
date_default_timezone_set('America/Mexico_City');
//Abrir la base de datos
$servername = "localhost";
$username = "id17851529_alumnos";
$password = "AlumnosAB123$";
$dbname = "id17851529_misdatos";
//obtener las variables que son enviadas por el dispositivo IoT
if($_GET)
{
```

```
$temperatura = $_GET["temperatura"];
$presion = $_GET["presion"];
$altitud = $_GET["altitud"];
$date = new DateTime("now", new DateTimeZone('America/New_
York') );
$fecha = $date->format('Y-m-d H:i:s');
echo $date->format('Y-m-d H:i:s');
// Conectarse a MySql
$conn = new mysqli($servername, $username, $password, $db-
name);
// Revisar si se puede conectar a la base de datos
if ($conn->connect_error) {
die("La conexión fallo: " . $conn->connect_error);
}

//insertar los valores de las variables en la tabla de la base de datos
$sql = "INSERT INTO estacion (temperatura, presion, altitud,fecha)
VALUES ($temperatura, $presion, $altitud,'$fecha')";

if ($conn->query($sql) = = = TRUE) {
echo "Nuevo registro añadido con exito";
} else {
echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
}
?>
```

Aquí se utilizó un servidor gratuito de hospedaje llamado 000webhost ubicado en la dirección https://www.000webhost.com/ en el cual se almacenó el programa en PHP y la base de datos de MySQL.

3. Ahora veremos la programación necesaria del NodeMCU8266 para enviar los datos de la temperatura, presión barométrica y altitud hacia el servidor de base de datos de MySQL que se encuentra en Internet, seleccionando la opción de Archivo->Nuevo y luego capturando el siguiente programa:

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <SFE_BMP180.h>
#include <Wire.h>

//Asignar los valores del nombre del punto de acceso y
//contraseña
const char* ssid = "AP-PTCs";
const char* password = «tc-ptcs2014»;
const uint16_t port = 443;
//definir al servidor que se va a conectar
const char * host = "pmoviles1.000webhostapp.com";
//Definir el objeto para leer las variable sensadas el BMP180
SFE_BMP180 pressure;

#define ALTITUDE 500.0 // Altitud base para calibrar el sensor a Co-
lima

void setup()
{
Serial.begin(115200);
delay(10);

// Conectarse por WiFi a Internet
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED)
delay(500);
```

```
Serial.println("Conectado a Internet");

// Inicialice el sensor (es importante obtener los valores de calibración almacenados en el dispositivo).

if (pressure.begin())
Serial.println("BMP180 se inicializo con exito");
else
{
Serial.println("BMP180 fallo la inicialización\n\n");
while(1); // Quedarse pausado.
}
}

void loop()
{

//Inicializar los objetos para conectarse por HTTP
BearSSL::WiFiClientSecure client;
client.setInsecure();
HTTPClient http;

char status;
double T,P,p0,a;

/*
Haz un bucle aquí obteniendo lecturas de presión cada 10 segundos.
Si desea una presión compensada por el nivel del mar, como se usa en los informes meteorológicos,
necesitará saber la altitud a la que se toman sus mediciones.
Estamos usando una constante llamada ALTITUDE en este boceto:
*/
Serial.println();
Serial.print("altitud proporcionada: ");
```

```
Serial.print(ALTITUDE,0);
Serial.print(" metros, ");
Serial.print(ALTITUDE*3.28084,0);
Serial.println(" pies");

/*
Si desea medir la altitud, y no la presión, en su lugar necesitará
proporcionar una presión de referencia conocida. Esto se muestra al
final del boceto.
Primero debe obtener una medición de temperatura para realizar una
lectura de presión.

Inicie una medición de temperatura:
Si la solicitud se realiza correctamente, se devuelve el número de ms
a esperar.
Si la solicitud no se realiza correctamente, se devuelve 0.
*/

status = pressure.startTemperature();
if (status != 0)
{
// Espere a que se complete la medición:
delay(status);

//Recupere la medición de temperatura completada:
//Tenga en cuenta que la medición se almacena en la variable T.
//La función devuelve 1 si tiene éxito, 0 si falla.

status = pressure.getTemperature(T);
if (status != 0)
{
// Imprima la medición:
Serial.print("temperatura: ");
Serial.print(T,2);
```

```
Serial.print(« grados C, «);
Serial.print((9.0/5.0)*T+32.0,2);
Serial.println(“ grados F”);

/*
Inicie una medición de presión:
El parámetro es la configuración de sobremuestreo, de 0 a 3 (resolución más alta, espera más larga).
Si la solicitud se realiza correctamente, se devuelve el número de ms a esperar.
Si la solicitud no se realiza correctamente, se devuelve 0.
*/

status = pressure.startPressure(3);
if (status != 0)
{
// Espere a que se complete la medición:
delay(status);

/*
Recupere la medición de presión completada:
Tenga en cuenta que la medición se almacena en la variable P.
Tenga en cuenta también que la función requiere la medición de temperatura previa (T).
(Si la temperatura es estable, puede realizar una medición de temperatura para una serie de mediciones de presión).
La función devuelve 1 si tiene éxito, 0 si falla.
*/

status = pressure.getPressure(P,T);
if (status != 0)
{
// Imprima la medición.:
Serial.print(“presión absoluta: “);
```

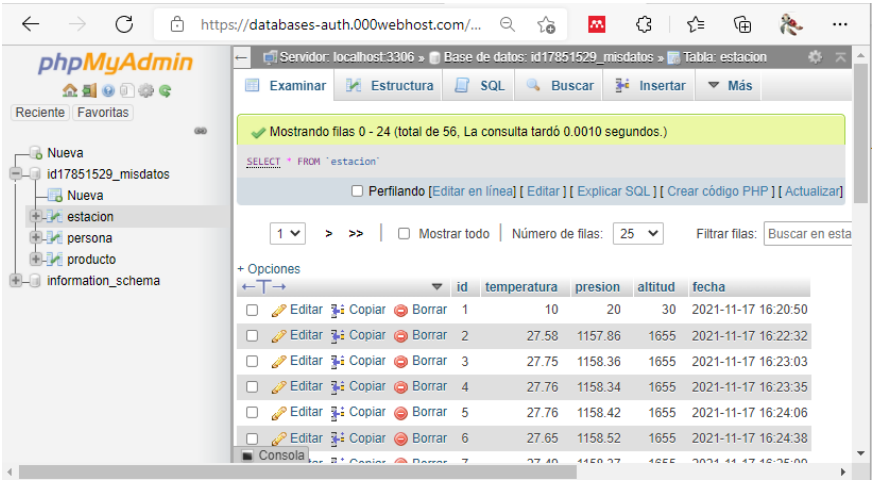
```
Serial.print(P,2);
Serial.print(" mb, ");
Serial.print(P*0.0295333727,2);
Serial.println(" inHg");
/*
El sensor de presión devuelve la presión absoluta, que varía con la altitud.
Para eliminar los efectos de la altitud, utilice la función del nivel del mar y su altitud actual.
Este número se usa comúnmente en los informes meteorológicos.
Parámetros: P = presión absoluta en mb, ALTITUD = altitud actual en m.
Resultado: p0 = presión compensada por el nivel del mar en mb
*/

p0 = pressure.sealevel(P,ALTITUDE); // estamos a 500 metros (Colima, Col.)
Serial.print("rpresión relativa (nivel del mar): ");
Serial.print(p0,2);
Serial.print(" mb, ");
Serial.print(p0*0.0295333727,2);
Serial.println(" inHg");
/*
Por otro lado, si desea determinar su altitud a partir de la lectura de presión,
utilice la función de altitud junto con una presión de referencia (nivel del mar u otra).
Parámetros: P = presión absoluta en mb, p0 = presión basal en mb.
Resultado: a = altitud en m.
*/
a = pressure.altitude(P,p0);
Serial.print("altitud calculada: ");
Serial.print(a,0);
Serial.print(" metros, ");
```

```
Serial.print(a*3.28084,0);
Serial.println(" pies");
}
else Serial.println("Error al recuperar la medición de presión\n");
}
else Serial.println("error de medición de la presión de inicio\n");
}
else Serial.println("Error al recuperar la medición de temperatura\n");
}
else Serial.println("error al iniciar la medición de la temperatura\n");

//Agregar los datos al servidor de Mysql vía http por el método GET
String path = "/iot/agregar.php?temperatura=" + String(T) + "&pre-
sion=" + String(p0) + "&altitud=" + String(a);
if (http.begin(client, host, port, path)) //Iniciar conexión
{
Serial.print("[HTTP] GET...\n");
int httpCode = http.GET(); // Realizar petición

if (httpCode > 0) {
Serial.printf("[HTTP] GET... codigo: %d\n", httpCode);

if (httpCode == HTTP_CODE_OK || httpCode == HTTP_CODE_
MOVED_PERMANENTLY) {
String payload = http.getString(); // Obtener respuesta
Serial.println(payload); // Mostrar respuesta por serial
}
}
else {
Serial.printf("[HTTP] GET... fallo, error: %s\n", http.errorTo-
String(httpCode).c_str());
}

http.end();
```

```
}
else {
Serial.printf("[HTTP} No se puede conectar\n");
}
//
delay(30000);
}
```

4. Para guardar el programa en la computadora debes seleccionar del menú Archivo->Salvar y asignarle el nombre al *sketch* o presionar las teclas Ctrl más la tecla S (Ctrl + S).
5. Ahora debes compilar el programa seleccionando del menú Programa->Verificar/Compilar o presionando las teclas Ctrl más la tecla R (Ctrl + R), otra opción es desde el IDE seleccionar el botón que tiene una flecha .
6. Si no hubo ningún problema al compilarlo, ya puedes subir tu programa a la placa Arduino Uno y ver el resultado, para ello selecciona del menú Programa->Subir o presiona las teclas Ctrl más la tecla U (Ctrl + U), otra opción es desde el IDE seleccionar el botón que tiene una flecha hacia la derecha , después de ello podrás ver que el sistema se conectará a Internet e iniciará el envío de los datos hacia la base de MySQL.
7. Puedes ir a tu servidor de base de datos de MySQL donde verás reflejado el envío de los datos de temperatura, presión y altitud en una tabla llamada estación, como se muestra en la figura 3.

Figura 3. Extracto de datos enviados a MySQL desde el NodeMCU



Fuente: Elaboración propia, captura de pantalla.

Ya solo resta que montes tu práctica y la pongas a prueba, para que veas los resultados y realices un análisis de lo mostrado y lo aprendido.

Práctica: escribir y leer datos en una base de Firebase desde un NodeMCU ESP8266 conectado a Internet

Objetivo general

El objetivo de esta práctica es escribir y leer datos utilizando la base en tiempo real Firebase, para ello habrá que crear la base de datos en esta herramienta, conectar a Internet el NodeMCU ESP8266, configurar la conexión del NodeMCU para que se conecte a la base de datos y entonces escribir datos en ella generados de manera aleatoria, además de declarar una variable llamada “LED” para colocarle valores en Firebase y poderlos leer desde el módulo ESP8266.

Materiales

1. Un módulo NodeMCU ESP8266.
2. Un cable USB de impresora.
3. Una computadora.
4. Arduino IDE.

Firebase

Es una herramienta desarrollada por la empresa Google para construir aplicaciones principalmente para Web y móviles; se encuentra ubicada en la nube como parte de los servicios de Google Cloud Platform, lo cual permite que los proyectos tengan una sincronización automática de los datos de los proyectos, sin necesidad de tener que escribir programas de lógica para las conexiones de forma compleja, aunado a estas ventajas tenemos la principal: está en el *back-end* o servidor, que es tener una base de datos en tiempo real, organizada como un árbol de tipo JSON (un formato de texto sencillo para el intercambio de datos) y que, mediante una API, permite que los datos de nuestras aplicaciones se sincronicen y almacenen en la base de datos de Firebase que se encuentra en la nube. En IoT sus ventajas son enormes, la primera de ella es detectar la presencia de los dispositivos que estén activos y, todavía mejor, recibir información de ellos en tiempo real o interactuar con el dispositivo para accionar otros objetos en su momento (Prasetyawan et al., 2021).

Procedimiento

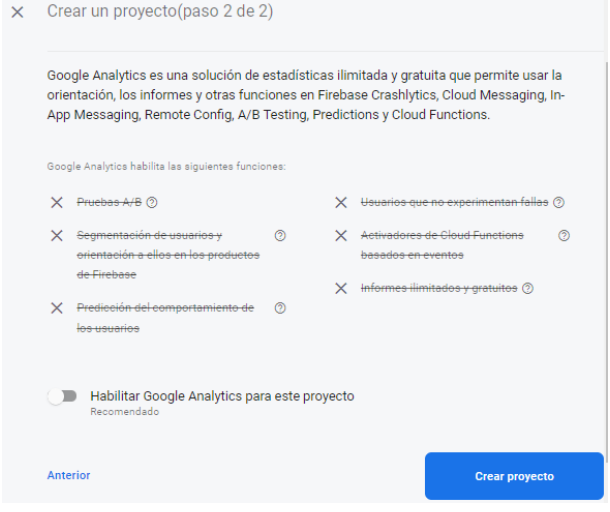
1. Es necesario configurar el Arduino IDE para que pueda trabajar con el módulo o tarjeta NodeMCU ESP8266.
2. Ahora se crea la base de datos en Firebase donde se almacenarán los datos, para ello se genera un nuevo proyecto:
   a. Ir a Firebase e iniciar sesión con una cuenta de Google.
   b. Hacer clic en Comenzar y luego en Agregar proyecto para crear un nuevo proyecto.
   c. Asignar un nombre a tu proyecto, por ejemplo: cursoiot, como se sugiere en la figura 1.

Figura 1. Cómo crear nuevo proyecto Firebase

Fuente: Elaboración propia, captura de pantalla.

d. Deshabilitar la opción "Habilitar Google Analytics", ya que no es necesaria y hacer clic en "Crear proyecto", como se ilustra en la figura 2.

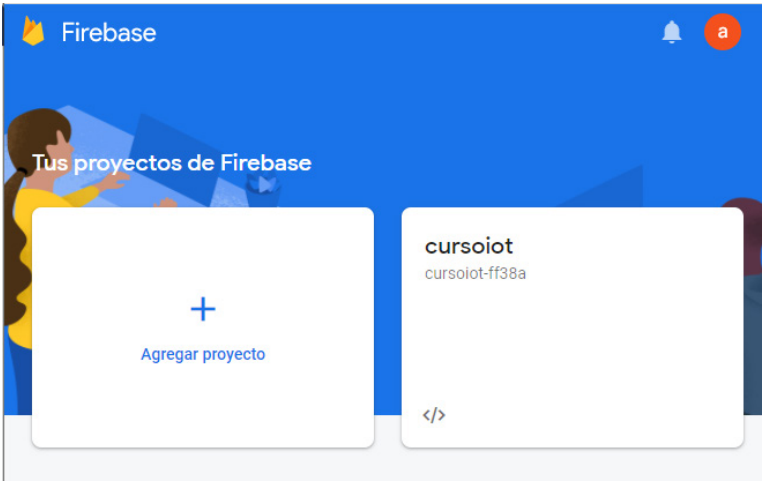
Figura 2. Deshabilitar Google Analytics



Fuente: Elaboración propia, captura de pantalla.

e. A Firebase le llevará unos segundos configurar el proyecto. Luego, hacer clic en Continuar cuando esté listo, como se indica en la figura 3.
f. Serás redirigido a la página de la consola de tu proyecto.

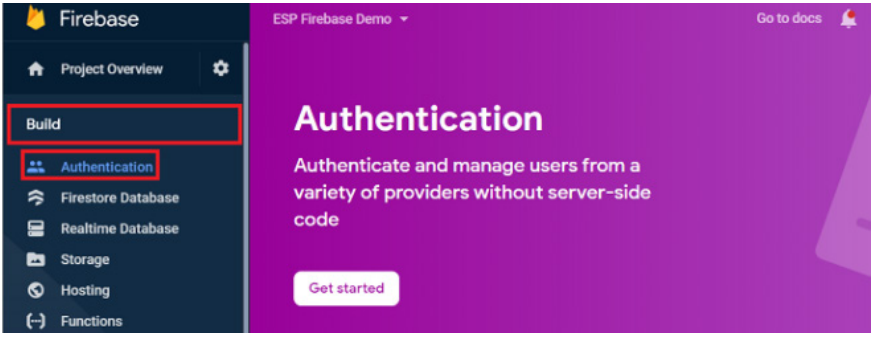
Figura 3. Vista de consola Firebase



Fuente: Elaboración propia, captura de pantalla.

g. Ahora se debe asignar el método de autenticación, la mayoría de las aplicaciones necesitan conocer la identidad de un usuario. Es decir, se encarga de iniciar sesión e identificar a los usuarios (en este caso, el ESP8266). Conocer la identidad de un usuario permite que una aplicación guarde de forma segura los datos de dicho usuario en la nube y brinde la misma experiencia personalizada en todos sus dispositivos. En la barra lateral izquierda, haga clic en Autenticación y luego en Comenzar, como se muestra en la figura 4.

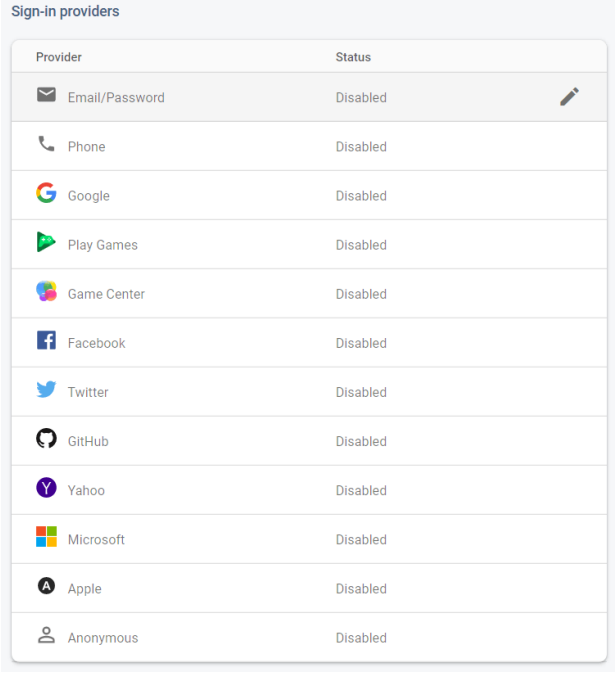
Figura 4. Ubicación de opción Autenticación



Fuente: Elaboración propia, captura de pantalla.

h. Hay varios métodos de autenticación como correo electrónico y contraseña, cuenta de Google, cuenta de Facebook y otros, tal como se aprecia en la figura 5.

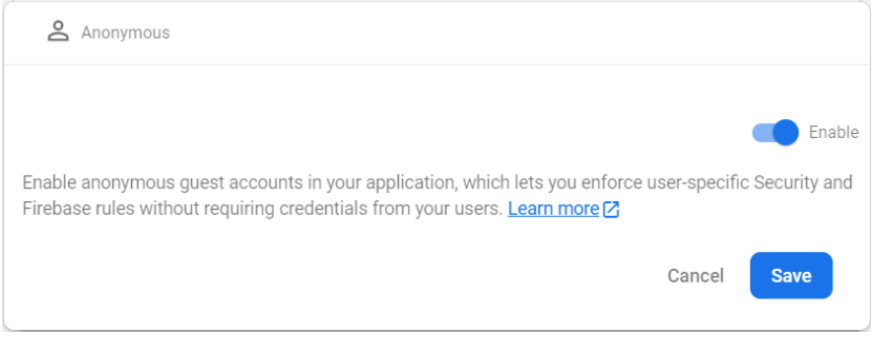
Figura 5. Métodos de autenticación en Firebase



Fuente: Elaboración propia, captura de pantalla.

i. Para fines de prueba, se puede seleccionar el usuario anónimo (requiere autenticación sin requerir que los usuarios inicien sesión primero, creando cuentas anónimas temporales) como se muestra en la figura 6. Habilitar esa opción y dar clic en Save.

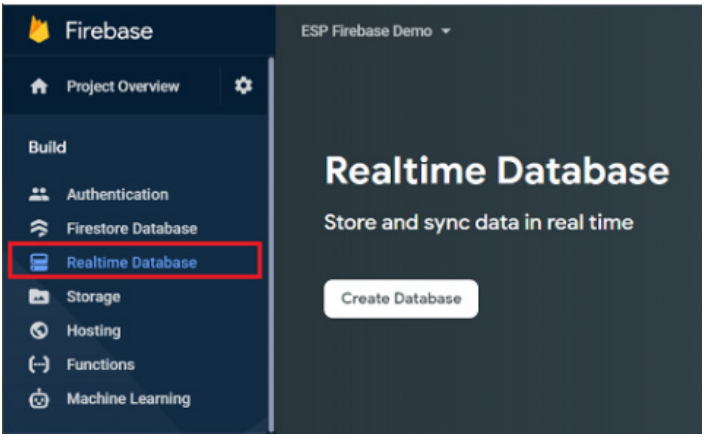
Figura 6. Pantalla de autenticación anónima



Fuente: Elaboración propia, captura de pantalla.

j. En la barra lateral izquierda, dar clic en *Realtime Database* y luego dar clic en Crear base de datos, como se muestra en la figura 7.

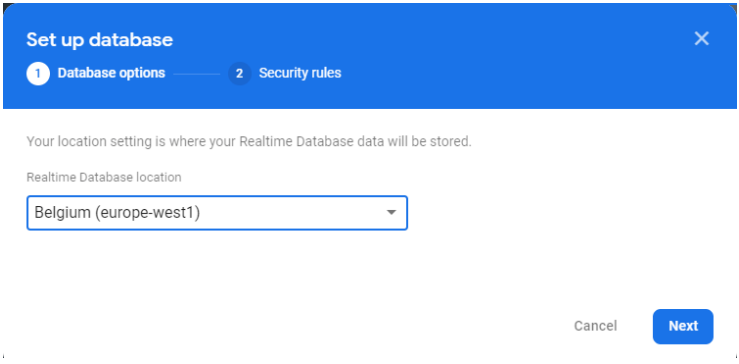
Figura 7. Creación de base de datos



Fuente: Elaboración propia, captura de pantalla.

k. Seleccionar la ubicación de tu base de datos. Debe ser el más cercano a tu ubicación, ver figura 8.

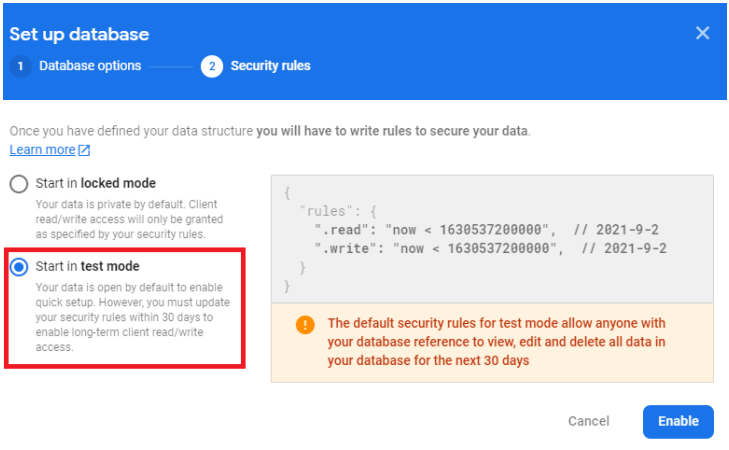
Figura 8. Pantalla de selección de servidor



Fuente: Elaboración propia, captura de pantalla.

l. Configurar las reglas de seguridad para tu base de datos. Para realizar pruebas, seleccionar Iniciar en modo de prueba, como se presenta en la figura 9.

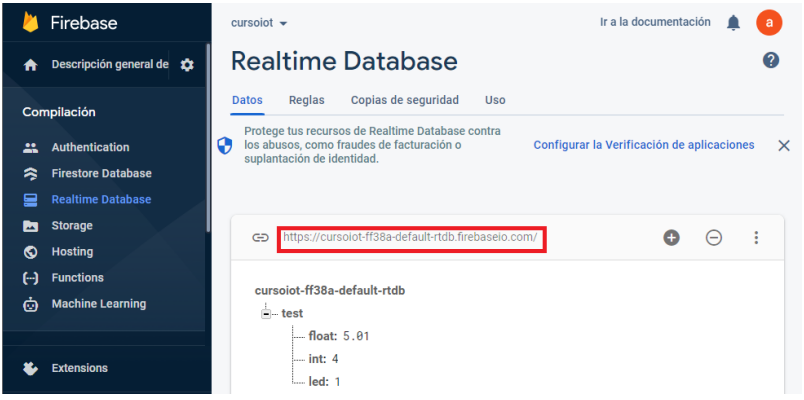
Figura 9. Selección del modo de prueba



Fuente: Elaboración propia, captura de pantalla.

m. Tu base de datos ahora está creada. Debes copiar y guardar la URL de la base de datos, resaltada en la figura 10, porque se necesitará más adelante en el código ESP8266.

Figura 10. Ejemplo de captura de URL

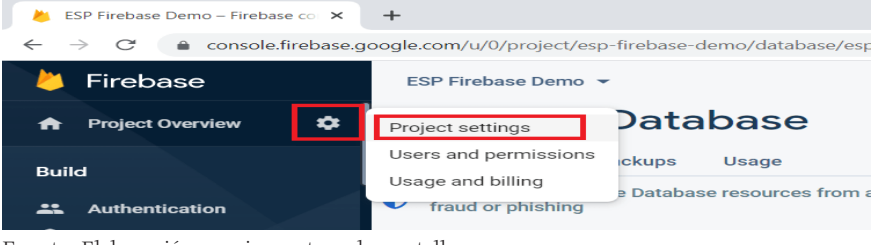


Fuente: Elaboración propia, captura de pantalla.

n. Obtener la clave de la API del proyecto necesaria para que ESP8266 se conecte a la base de datos de Firebase.

Para obtener la clave API de su proyecto, en la barra lateral izquierda, haga clic en Configuración del proyecto, como se muestra en la figura 11.

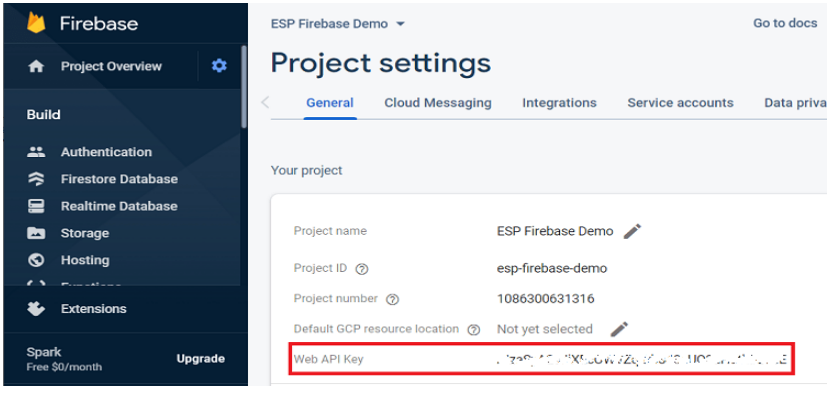
Figura 11. Selección de configuración para clave API



Fuente: Elaboración propia, captura de pantalla.

Copia la clave de API en un lugar seguro porque la necesitarás más adelante, como se muestra en la figura 12.

Figura 12. Captura de web API Key

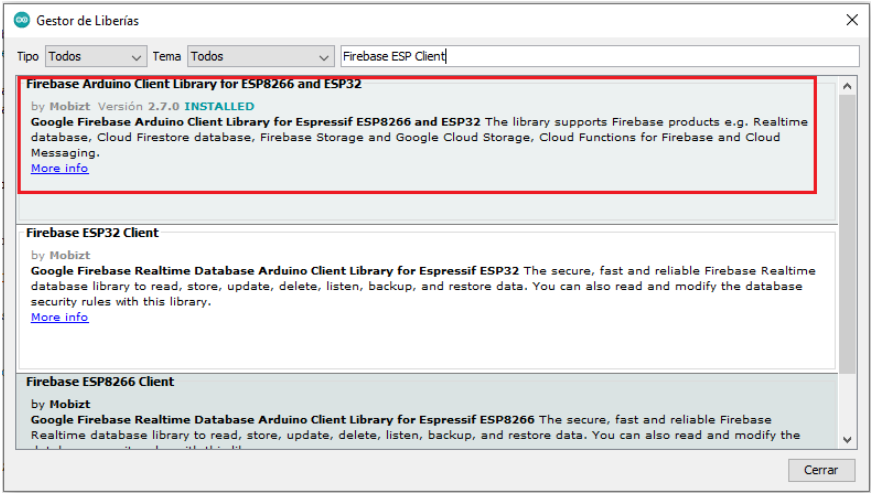


Fuente: Elaboración propia, captura de pantalla.

Y ahora sí, se tiene todo listo para conectar el ESP8266 con la base de datos de Firebase.

3. Ahora que ya creaste la Firebase *Realtime Database*, aprenderás a conectar el ESP8266 con la base de datos. Hay una biblioteca con muchos ejemplos para usar Firebase con las placas ESP8266 y ESP32: la biblioteca Firebase-ESP-Client. Hay que entrar al Arduino IDE y seguir los siguientes pasos para instalar la biblioteca, ver figura 13:
    a. Ir a Programa-> Incluir librería-> Administrar bibliotecas…
    b. Buscar Firebase ESP Client e instar Firebase Arduino Client Library para ESP8266 y ESP32 de Mobitz.

Figura 13. Ejemplo para agregar librería



Fuente: Elaboración propia, captura de pantalla.

Ahora, ya está todo listo para comenzar a programar la placa ESP8266 a fin de interactuar con la base de datos.

4. Ahora veremos la programación necesaria del NodeMCU8266 para enviar variables *Int*, *Float* hacia el servidor de base de datos de Firebase que se encuentra en Internet, seleccionando la opción de Archivo->Nuevo y luego capturando el siguiente programa:

```
#include <Arduino.h>
#if defined(ESP32)
  #include <WiFi.h>
#elif defined(ESP8266)
  #include <ESP8266WiFi.h>
#endif
#include <Firebase_ESP_Client.h>

//Proporcionar la información del proceso de generación de tokens.
#include "addons/TokenHelper.h"
```

```
//Proporcione la información de impresión de la carga útil de RTDB
y otras funciones auxiliares.
#include "addons/RTDBHelper.h"

// Inserte sus credenciales de red
#define WIFI_SSID "AP-PTCs"
#define WIFI_PASSWORD "tc-ptcs2014"

// Insertar clave de API del proyecto de Firebase
#define API_KEY "AIzaSyDCPhyOKKBHkQbue6PdFsj53SICJ_SigtM"

// Insertar URL de RTDB (Real Time Database) de firebase
#define DATABASE_URL "https://cursoiot-ff38a-default-rtdb.fireba-
seio.com/"

//Definir el objeto de datos de Firebase
FirebaseData fbdo;

FirebaseAuth auth;
FirebaseConfig config;

unsigned long sendDataPrevMillis = 0;
int count = 0;
bool signupOK = false;
int intValue;

void setup(){
  Serial.begin(115200);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Conexión a Wi-Fi");
  while (WiFi.status() != WL_CONNECTED){
    Serial.print(".");
    delay(300);
  }
```

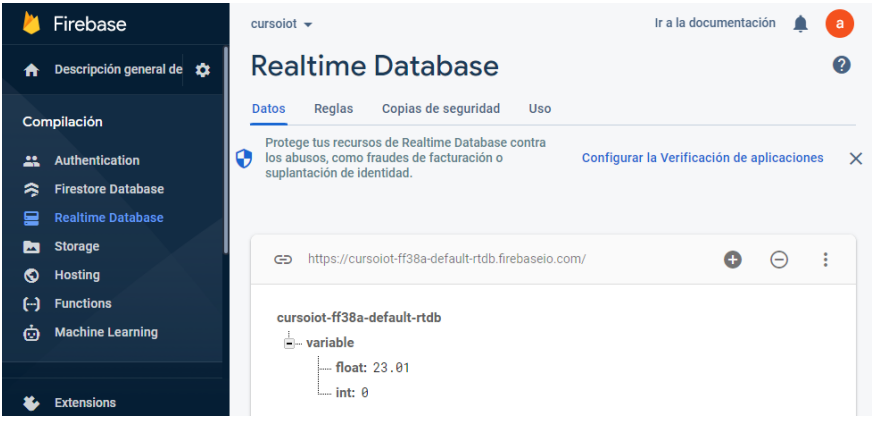
```
  Serial.println();
  Serial.print("Conectado con IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();

  /* Asignar la clave api (obligatorio) */
  conFiguraapi_key = API_KEY;

  /* Asignar la URL de RTDB (obligatorio) */
  conFiguradatabase_url = DATABASE_URL;

  /* Inscribirse y autenticarse */
  if (Firebase.signUp(&config, &auth, "", "")){
    Serial.println("ok");
    signupOK = true;
  }
  else{
    Serial.printf("%s\n", conFigurasigner.signupError.message.c_str());
  }

  /* Asignar la función de devolución de llamada para la tarea de generación de tokens de larga duración */
  conFiguratoken_status_callback = tokenStatusCallback; //ver complementos / TokenHelper.h

  Firebase.begin(&config, &auth);
  Firebase.reconnectWiFi(true);
}

void loop(){
  //Guardar datos en la base de datos de firebase
  if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis > 15000 || sendDataPrevMillis == 0)){
    sendDataPrevMillis = millis();
```

```
  // Escriba un número Int en la ruta de la base de datos varible/int
  if (Firebase.RTDB.setInt(&fbdo, "variable/int", count)){
    Serial.println("APROBADO");
    Serial.println("RUTA: " + fbdo.dataPath());
    Serial.println("TIPO: " + fbdo.dataType());
  }
  else {
    Serial.println("FALLO");
    Serial.println("RASON: " + fbdo.errorReason());
  }
  count++;

  // Write an Float number on the database path variable/float
    if (Firebase.RTDB.setFloat(&fbdo, "variable/float", 0.01 + ran-
dom(0,100))){
    Serial.println("APROBADO");
    Serial.println("RUTA: " + fbdo.dataPath());
    Serial.println("TIPO: " + fbdo.dataType());
  }
  else {
    Serial.println("FALLO");
    Serial.println("RASON: " + fbdo.errorReason());
  }
 }
//Leer el valor de la variable LED desde firebase

if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis
> 15000 || sendDataPrevMillis == 0)) {
  sendDataPrevMillis = millis();
  if (Firebase.RTDB.getInt(&fbdo, "/variable/led")) {
    if (fbdo.dataType() == "int") {
      intValue = fbdo.intData();
      Serial.println("Valor de la variable led=");
      Serial.println(intValue);
```

```
    }
   }
   else {
    Serial.println(fbdo.errorReason());
   }
  }
}
```

5. Para guardar el programa en la computadora debes seleccionar del menú Archivo->Salvar y asignar el nombre al *sketch* o presiona las teclas Ctrl más la tecla S (Ctrl + S).
6. Ahora debes compilar el programa seleccionando del menú Programa->Verificar/Compilar o presiona las teclas Ctrl más la tecla R (Ctrl + R), otra opción es desde el IDE seleccionar el botón que tiene una flecha .
7. Si no hubo ningún problema al compilarlo, ya puedes subir tu programa a la placa Arduino Uno y ver el resultado, para ello, selecciona del menú Programa->Subir o presiona las teclas Ctrl más la tecla U (Ctrl + U), otra opción es desde el IDE seleccionar el botón que tiene una flecha hacia la derecha , después de ello el sistema se conectará a Internet y empezará a enviar los datos hacia la base de datos de Firebase.
8. Puedes ir a tu servidor de base de datos de Firebase donde verás reflejada la creación de la categoría variable y las variables llamadas *Float* e *Int*, así como el envío de los datos generados por el programa que actualizaran las variables *Float*, *Int* como se muestra en la figura 14.

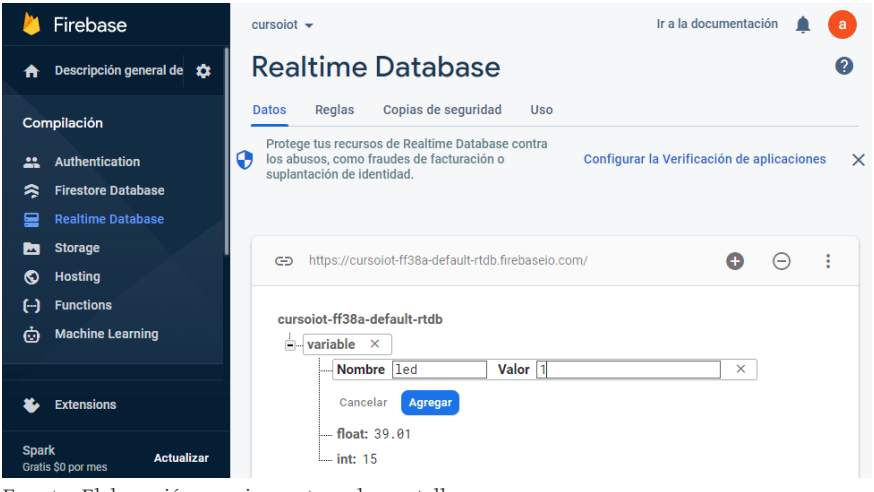
Figura 14. Vista de las variables creadas



Fuente: Elaboración propia, captura de pantalla.

Como podrás observar, no se encuentra la variable *LED*, esta se tendrá que crear agregándola, para ello hay que seleccionar el tema variable dar clic cuando aparezca el + y agregar la variable LED con el valor de 1, como se muestra en la figura 15.

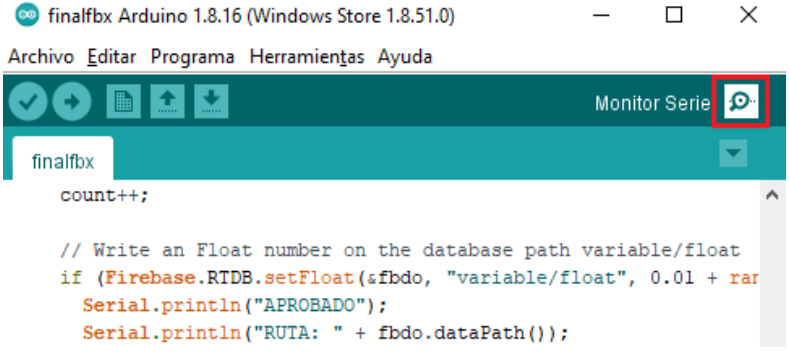
Figura 15. Agregar nueva variable



Fuente: Elaboración propia, captura de pantalla.

Y entonces se verá agregada a la colección llamada variable. Para ver que el módulo está recibiendo el valor de la variable LED es necesario ir al monitor de puerto serial del Arduino IDE, seleccionando este como se muestra en la Figura 16.

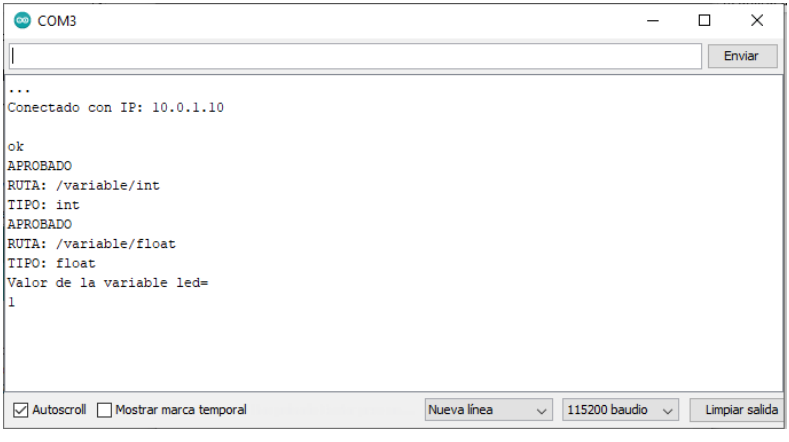
Figura 16. Selección del Monitor serial para ver los datos



Fuente: Elaboración propia, captura de pantalla.

Una vez que se activó el monitor serial se puede ver cómo se lee el valor de la variable LED y la muestra en este, como se aprecia en la figura 17.

Figura 17. Consola serial con resultados



Fuente: Elaboración propia, captura de pantalla.

Si se cambia el valor de la variable LED, esto se verá reflejado en el valor mostrado en el monitor serie que te arrojará el valor que tiene actualmente, con ese valor se puede accionar algo, por ejemplo, encender un LED, un foco, una bomba de agua, etc. Ya solo resta que se realice toda la práctica y se verifique el funcionamiento y el diseño de nuevos proyectos afines.

Conclusiones

Los ejercicios de adiestramiento presentados son un primer acercamiento para que los interesados tengan conocimientos más prácticos sobre la electrónica, conectividad, los protocolos de comunicación, e implementaciones del desarrollo en IoT en forma embebida. Todo ello es presentado de una manera lógica y puntual, incluyendo sugerencias y código, así como herramientas de diseño para que puedan accionar los dispositivos de interés en la realización de tareas específicas.

Referencias

Abdullah, E., Shuhaimi, N. I., Mohamad, R., Kamaluddin, N. A., Hidayat, N. M. y Burham, N. (2021). Development of real-time energy monitoring system and data log using NodeMCU ESP 8266 and MYSQL Database. https://www.researchgate.net/publication/356509140_Development_of_Real-Time_Energy_Monitoring_System_and_Data_Log_Using_NodeMCU_ESP_8266_and_MYSQL_Database

Almenara, J. A. (2020). *Proyecto fin de carrera Ingeniería de Telecomunicación.* Universidad de Sevilla.

Álvarez Carulla, A. (2021). *Comunicación de un módulo ESP32 con Ubidots mediante MQTT*. http://hdl.handle.net/2445/181905

Boyes, H., Hallaq, B., Cunningham, J. y Watson, T. (2018). The industrial internet of things (IIoT): an analysis framework. *Computers in Industry,* 101, 1-12.

Geek Factory. (2020). *Conectar un led a tu Arduino*. https://www.geekfactory.mx/tutoriales-arduino/conectar-un-led-a-tu-arduino/

Hwang, K., Lee, J. M., Jung, I. H. y Lee, D.-H. (2019). Modification of mosquitto broker for delivery of urgent MQTT message. En *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)* (pp. 166-167).

Ilham, A. (2020). *Ancang bangun stasiun cuaca menggunakan dht22, bmp180, nodemcu dengan arduino ide melalui blynk server*. Nusa Putra University.

Junizan, N. A., Razak, A. A., Balakrishnan, B. y Othman, W. (2019). Design and implementation of automatic room temperature controlled fan using Arduino Uno and LM35 heat sensor. *International Journal of Engineering Creativity & Innovation, 1*(2), pp. 8-14.

Mishra, B. y Kertesz, A. (2020). The use of MQTT in M2M and IoT systems: a survey. *IEEE Access*, 8, 201071-201086.

Novillo-Vicuña, J., Rojas, D. H., Olivo, B. M., Ríos, J. M. y Villavicencio, O. C. (2018). Arduino y el Internet de las cosas. *Ciencias, 45*(3). http://dx.doi.org/10.17993/IngyTec.2018.45

Parihar, Y. S. (2019). Internet of things and Nodemcu. *Journal of Emerging Technologies and Innovative Research, 6*(6), 1085. http://dx.doi.org/10.13140/RG.2.2.34456.75525

Peña, C. (2020). *Arduino IDE: Domina la programación y controla la placa*. RedUsers.

Perkasa, R., Wahyuni, R., Melyanti, R. e Irawan, Y. (2021). Light control using human body temperature based on Arduino Uno and PIR (Passive Infrared Receiver) Sensor. *Journal of Robotics and Control (JRC), 2*(4), pp. 307-310. https://journal.umy.ac.id/index.php/jrc/article/download/10182/5829

Prasetyawan, P., Samsugi, S. y Prabowo, R. (2021). Internet of things menggunakan Firebase dan Nodemcu untuk Helm Pintar. *Jurnal ELTIKOM: Jurnal Teknik Elektro, Teknologi Informasi Dan Komputer, 5*(1), pp. 32-39.

Naylamp mechatronics (2020). *Usando ESP8266 con el IDE de Arduino*. https://naylampmechatronics.com/blog/56_usando-esp8266-con-el-ide-de-arduino.html

Valderrama, J. y Brea, E. (2020). ESP8266: Un microcontrolador para el Internet de las cosas. En *Jornadas de investigación.* Universidad Central de Venezuela, Tech. Rep. http://www.ing.ucv.ve/jifi2018/documentos/electronica/ERT-006.pdf

Vergara, J. E. (2021). Ingenieros industriales desarrollando proyectos con tecnologías IOT y Arduino – ¿Qué se necesita, ¿cómo iniciar? *Gestión Competitividad e Innovación, 9*(1), pp. 1-10. https://pca.edu.co/editorial/revistas/index.php/gci/article/view/131

Reseñas curriculares

Armando Román Gallardo

Profesor-investigador de tiempo completo en la Facultad de Telemática de la Universidad de Colima. Su formación académica reside en la Ingeniería en Sistemas Computacionales y la Maestría en Ciencias Computacionales por la Universidad de Colima; y el Doctorado en Educación por la Universidad de Baja California. Sus intereses de investigación se centran en los procesos de desarrollo de *software*.

Arturo Cano Rueda

Graduado de Ingeniería de Software en la Universidad de Colima, actualmente se dedica al Desarrollo Web FullStack; sus principales intereses se encuentran en la gamificación de los procesos y las aplicaciones en tiempo real.

Carlos Alberto Flores Cortés

Profesor-investigador de la Facultad de Telemática de la Universidad de Colima. Ingeniero en comunicaciones y electrónica. Maestro en Telemática por la Universidad de Colima y Doctor en Computación por la Universidad de Lancaster (Inglaterra). Su línea de investigación se relaciona con los sistemas distribuidos.

Erika Margarita Ramos Michel

Profesora-investigadora de tiempo completo en la Facultad de Telemática de la Universidad de Colima. Es Ingeniera en Comunicaciones y Electrónica y Maestra en Ciencias área Telemática por la Universidad de Colima, con Doctorado en Ciencias de la Computación por el Centro de Investigación Científica y de Educación Superior de Ensenada-Baja California (CICESE). Sus intereses de investigación incluyen procesamiento de señales e imágenes, tecnología educativa e inteligencia computacional aplicada.

Jorge Rafael Gutiérrez Pulido

Profesor-investigador de tiempo completo en la Facultad de Telemática de la Universidad de Colima. Su formación académica reside en la Licenciatura en Informática y la Maestría en Telemática por la Universidad de Colima, y el Doctorado en Ciencias Computacionales por la Universidad de Nottingham (Reino Unido). Sus intereses de investigación son el web semántico, el aprendizaje artificial de ontologías, el minado de datos, el descubrimiento del conocimiento, *self organizing maps*, la recuperación semántica de la información, la visualización de datos, las bibliotecas digitales, y la vulcanología sísmica.

José Román Herrera Morales

Profesor-Investigador de tiempo completo en la Facultad de Telemática de la Universidad de Colima. Cursó la carrera de Ingeniería en Comunicaciones y Electrónica, y la Maestría en Ciencias, área Telemática, por la Universidad de Colima. Realizó su Doctorado en Tecnologías de la Información en la Universidad de Guadalajara. Sus intereses de investigación incluyen los sistemas de búsqueda y recuperación de información, los sistemas inteligentes, la tecnología web, minería y las bases de datos.

Juan Manuel Ramírez Alcaraz

Profesor-investigador de tiempo completo en la Facultad de Telemática de la Universidad de Colima. Obtuvo el grado de Doctor en Ciencias de la Computación por el Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE), en Ensenada Baja California. Recibió el grado de Maestría en Telemática de parte de la Universidad de Colima y el título de Ingeniero en Sistemas Computacionales de parte de la misma institución. Sus áreas de interés se enfocan en clusters, grid y cloud computing, calendarización, paralelismo y virtualización.

Margarita Glenda Mayoral Baldivia

Licenciada en Informática y Maestra en Ciencias área Telemática por la Universidad de Colima. Actualmente se desempeña como Profesora Investigadora de Tiempo Completo en la Facultad de Telemática, sus áreas de interés son el Cómputo Móvil, las redes y las telecomunicaciones

María Andrade-Aréchiga

Se desempeña como profesora-investigadora de tiempo completo en la Facultad de Telemática de la Universidad de Colima. Tiene la Maestría en Educación con especialidad en Matemáticas por el Instituto Tecnológico y de Estudios Superiores de Monterrey y el doctorado en Ciencias de la Computación por el Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE). Sus intereses de investigación se centran en el desarrollo, implementación y evaluación de ambientes de aprendizaje, Tecnología Educativa y Matemáticas. Cuenta con la distinción por el Sistema Nacional de Investigadores (SNI), nivel 1.

María Guadalupe Álvarez Negrete

Es Maestra en Tecnologías de Internet por la Universidad de Colima, en la Facultad de Telemática. Además, tiene estudios de Ingeniería en Telemática por parte de la misma institución. Actualmente es administradora de la WLAN Universitaria y docente por horas en la Facultad de Telemática de la Universidad de Colima. Sus áreas de interés se enfocan en Redes Inalámbricas, IoT, Sistemas Operativos y el área de innovación en la educación.

Omar Álvarez Cárdenas

Profesor-investigador de tiempo completo adscrito a la Facultad de Telemática en la Universidad de Colima. El profesor Álvarez es Ingeniero en Comunicaciones y Electrónica con una Maestría en Ciencias Área Telemática y un Doctorado en Educación. Sus investigaciones están relacionadas con las redes inalámbricas, redes de sensores, laboratorios remotos, cómputo móvil y tecnología educativa.

Pedro Damián Reyes

Profesor-investigador de tiempo completo en la Facultad de Telemática de la Universidad de Colima. Es Ingeniero en Sistemas Computacionales por el Instituto Tecnológico de Colima, Maestro en Ciencias área Telemática por la Universidad de Colima y Doctor en Ciencias de la Computación por el Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE). Sus trabajos de investigación se encuentran enfocados en al desarrollo de aplicación conscientes del contexto y realidad aumentada.

Ricardo Acosta Díaz

Profesor-investigador de tiempo completo en la Facultad de Telemática de la Universidad de Colima, donde realiza investigación sobre "Ambientes Inteligentes". Es doctor en Educación, Maestro en Ciencias de la Computación e Ingeniero en Sistemas Computacionales. Fue consultor de la Organización de Estados Americanos en el Desarrollo del Portal Educativo de las Américas. Realiza un Postdoctorado en currículum, innovación pedagógica y formación con enfoque en desarrollo de aplicaciones móviles que utilizan la gamificación como estrategia didáctica.

Raúl Aquino Santos

Miembro del Sistema Nacional de Investigadores. Ha publicado seis libros, 12 capítulos de libro y más de 30 artículos en revistas internacionales. Sus áreas de interés se encuentran en el diseño, desarrollo e implementación de industria 4.0, IoT, ciudades inteligentes y gestión de desastres naturales.

Sara Sandoval Carrillo

Profesora-investigadora de tiempo completo en la Facultad de Telemática de la Universidad de Colima. Su formación académica reside en la Ingeniería en Telemática y la Maestría en Telemática por la Universidad de Colima. Sus intereses de investigación se basan en el desarrollo de *software* dirigido por modelos y las tecnologías de la información.

Internet de las cosas. Teoría y práctica, fue editado en la Dirección General de Publicaciones de la Universidad de Colima, avenida Universidad 333, Colima, Colima, México, www.ucol.mx. La edición digital se terminó en enero de 2023. En la composición tipográfica se utilizó la familia ITC Veljovick Book. El tamaño del libro es de 22.5 cm por 15 cm de ancho. Programa Editorial: Eréndira Cortés Ventura. Gestión Administrativa: María Inés Sandoval Venegas. Corrección: Leticia Bermúdez Aceves. Diseño de portada: Adriana Vázquez. Diseño de interiores: José Luis Ramírez Moreno. Cuidado de la edición: Eréndira Cortés Ventura.

En esta obra se abordan aspectos relacionados con el *hardware* y *software* que hacen posible el Internet de las cosas, desde sus componentes, conectividad, protocolo, plataformas y aprendizaje automático que lo potencian, incluyendo ejercicios prácticos que permitirán al lector realizar implementaciones de IoT.

Si bien hace poco más de 20 años se empleó el término Internet de las cosas por primera vez, hoy en día, esta tecnología está presente en muchas actividades diarias y en ámbitos tan diversos como la escuela, el hogar, la industria, el transporte, entre otros. La posibilidad de conectar una gran cantidad de dispositivos entre sí, permite obtener información relativamente de todo lo que nos rodea. Si, además, se cuenta con la suficiente capacidad de almacenamiento y procesamiento, así como con técnicas analíticas y estadísticas convenientes, resulta entendible cómo se puede potenciar el cuidado de la salud, mejorar las cadenas de suministro y producción, proteger el medio ambiente, desarrollar vehículos autónomos, robots que apoyen en tareas domésticas o semáforos para evitar embotellamientos. Sin embargo, el avance de esta tecnología también impone retos, entre ellos la seguridad de la información, las limitaciones de conectividad y velocidad, la estandarización o la sostenibilidad.

ISBN: 978-607-8814-44-2

UNIVERSIDAD DE COLIMA